



**Universidad de Valladolid**

**ESCUELA DE INGENIERÍA INFORMÁTICA  
DE VALLADOLID**

**Grado en Ingeniería Informática**

---

**Sistema de check-in biométrico**

**TouchPass**

---

**Alumno: Jorge Andrés Nieto**

**Tutor/a/es: César Pablo Gutiérrez Martínez**

**Jose Baro González**



# TouchPass

Jorge Andrés Nieto



# Índice general

Lista de figuras	VII
Lista de tablas	IX
Resumen	I
<b>I Memoria del Proyecto</b>	<b>1</b>
<b>1. Introducción</b>	<b>3</b>
1.1. Estado del arte . . . . .	3
1.2. Descripción de la solución . . . . .	4
1.3. Objetivos de la solución . . . . .	5
1.4. Objetivos académicos del TFG . . . . .	6
1.5. Objetivos SMART . . . . .	6
<b>2. Gestión de proyecto del TFG</b>	<b>9</b>
2.1. Resumen . . . . .	9
2.2. Metodologías . . . . .	9
2.3. Colaboración con tutores . . . . .	10
2.4. Planificación . . . . .	11
2.5. Recursos . . . . .	12
2.5.1. Recursos Materiales . . . . .	12
2.5.2. Recursos Humanos . . . . .	13

2.6.	Presupuesto . . . . .	13
2.6.1.	Estimación y monitorización de los recursos a utilizar . . . . .	14
2.6.2.	Presupuesto total . . . . .	15
2.7.	DAFO - Debilidades Amenazas Fortalezas y Oportunidades . . . . .	15
2.8.	Gestión de riesgos . . . . .	17
2.8.1.	RSG01 Ausencia temporal del estudiante por enfermedad . . . . .	17
2.8.2.	RSG02 Fallos técnicos en el equipo de trabajo del estudiante . . . . .	18
2.8.3.	RSG03 Mala estimación de tiempo y trabajo de algunas actividades . . . . .	19
2.8.4.	RSG04 Ausencia temporal del tutor por causas no planificadas . . . . .	20
<b>3.</b>	<b>Requisitos de la solución</b>	<b>21</b>
3.1.	Requisitos Funcionales . . . . .	21
3.1.1.	RFA01 Biometría . . . . .	21
3.1.2.	RFA02 Autenticación . . . . .	22
3.1.3.	RFA03 API Rest . . . . .	22
3.1.4.	RFA04 Comunicación dispositivo servidor . . . . .	23
3.2.	Requisitos No Funcionales . . . . .	23
3.2.1.	RNFA01 Raspberry . . . . .	23
3.2.2.	RNFA02 Tecnologías Servidor . . . . .	23
3.2.3.	RNFA03 Persistencia de datos . . . . .	24
3.2.4.	RNFA04 JWT . . . . .	24
3.2.5.	RNFA05 UUID . . . . .	24
3.2.6.	RNFA06 Estándares de calidad . . . . .	24
3.2.7.	RNFA07 Linter . . . . .	24
3.2.8.	RNFA08 Cobertura de pruebas . . . . .	24
3.2.9.	RNFA09 Repositorio . . . . .	24
3.2.10.	RNFA01 Raspberry . . . . .	25
3.2.11.	RNFA02 Tecnologías Servidor . . . . .	25
3.2.12.	RNFA03 Persistencia de datos . . . . .	25

---

3.2.13. RNFA04 JWT . . . . .	25
3.2.14. RNFA05 UUID . . . . .	25
3.2.15. RNFA06 Estándares de calidad . . . . .	25
3.2.16. RNFA07 Linter . . . . .	25
3.2.17. RNFA08 Cobertura de pruebas . . . . .	26
3.2.18. RNFA09 Repositorio . . . . .	26
<b>4. Modelo de dominio</b>	<b>27</b>
<b>5. Análisis</b>	<b>29</b>
5.1. Diagrama de casos de uso . . . . .	29
5.2. Casos de uso . . . . .	31
5.2.1. CU01 Grabado de huella . . . . .	31
5.2.2. CU02 Alta de usuario . . . . .	32
5.2.3. CU03 Borrado de usuario . . . . .	33
5.2.4. CU04 Edición de usuario . . . . .	34
5.2.5. CU05 Acceso de usuario . . . . .	35
5.3. Diagrama de clases de análisis . . . . .	35
5.4. Diagramas de secuencia . . . . .	37
<b>6. Diseño</b>	<b>41</b>
6.1. Diagrama de estados . . . . .	41
6.2. Diagrama de despliegue . . . . .	43
6.3. Diagrama de clases de diseño . . . . .	44
6.4. Diagrama de base de datos . . . . .	46
6.5. Tecnologías . . . . .	46
6.5.1. Nodejs . . . . .	46
6.5.2. JWT . . . . .	48
6.5.3. MongoDB . . . . .	48

<b>7. Desarrollo</b>	<b>51</b>
7.1. Descripción de la arquitectura . . . . .	51
7.2. Configuración del entorno . . . . .	52
7.3. Integración de la base de datos . . . . .	53
7.4. Desarrollo adicional . . . . .	54
<b>8. Pruebas y validación</b>	<b>55</b>
8.1. Pruebas automáticas . . . . .	56
8.2. Pruebas con usuarios finales . . . . .	56
8.2.1. Segunda iteración . . . . .	58
<b>9. Resultados</b>	<b>59</b>
9.1. Generales . . . . .	59
9.2. Requisitos . . . . .	60
9.3. Riesgos . . . . .	60
9.4. Calendario . . . . .	61
9.5. Presupuesto . . . . .	61
<b>10. Conclusiones</b>	<b>63</b>
10.1. Sigüientes pasos: . . . . .	64
<b>II Apéndices</b>	<b>65</b>
<b>A. Anexos</b>	<b>67</b>
A.1. Información complementaria . . . . .	67
A.1.1. Especificaciones sensor biométrico . . . . .	67
A.1.2. Especificaciones técnicas raspberry . . . . .	69
A.1.3. Interfaz gráfica . . . . .	69
A.1.4. Diseño de la pieza 3D . . . . .	72
A.1.5. Links Adicionales . . . . .	73

**Bibliografía**

**75**



# Índice de figuras

2.1. Calendario del proyecto . . . . .	14
2.2. Estimación y monitorización de los recursos . . . . .	14
2.3. Presupuesto final del proyecto . . . . .	15
2.4. Presupuesto final del proyecto . . . . .	16
4.1. Modelo de dominio . . . . .	28
5.1. Diagrama de casos de uso . . . . .	30
5.2. Diagrama de clases de análisis . . . . .	36
5.3. Registro de huella . . . . .	38
5.4. Acceso . . . . .	39
6.1. Diagrama de estados . . . . .	42
6.2. Diagrama de despliegue . . . . .	44
6.3. Clases de diseño . . . . .	45
6.4. Diagrama de base de datos . . . . .	46
A.1. Pantalla principal . . . . .	70
A.2. Pantalla de detección de huella . . . . .	70
A.3. Pantalla de introducción de nombre de usuario . . . . .	71
A.4. Vista superior del conjunto tapa-sensor-led . . . . .	72
A.5. Vista de las protecciones . . . . .	73



# Índice de cuadros

5.1. CU-01. Grabado de huella . . . . .	31
5.2. CU-02. Alta de usuario . . . . .	32
5.3. CU-03. Borrado de usuario . . . . .	33
5.4. CU-04. Edición de usuario . . . . .	34
5.5. CU-05. Acceso de usuario . . . . .	35



*Dedicado a  
todas esas personas  
que creen en mí*



# Agradecimientos

Quiero agradecer en primer lugar a mi familia (Papá, Mamá, Guille), que han estado para mí siempre y pese a todo cuando más lo he necesitado, y sin ellos no habría podido superar esta etapa de mi vida de la forma en que lo he hecho.

Por otro lado, a mi grupo de amigos del grado, cuyo nombre no se puede mencionar, que han hecho de estos años algo más que estudiar y aprobar exámenes, disfrutando cada tontería.

Y por último, aunque sea algo más reciente, también reservo parte de estos agradecimientos a María, por aguantarme y sacar lo mejor de mí cuando a veces ni yo mismo lo veo.

Muchas gracias a todos.



# Resumen

Este proyecto tiene como objetivo principal implementar un sistema de check-in que utilice tecnología biométrica para verificar la identidad de los empleados de una empresa de forma rápida y segura. La implementación de este sistema mejorará la eficiencia en el proceso de entrada y salida de los empleados en la oficina, reduciendo el tiempo y los esfuerzos necesarios para el control manual de acceso.

El sistema utiliza una Raspberry Pi como plataforma de hardware y un sensor biométrico de huellas dactilares para capturar la identidad de los empleados. La información biométrica capturada es procesada por una API REST basada en NodeJS que se encarga de verificar la identidad de los empleados y de registrar su entrada y salida en la oficina.

Este proyecto de sistema de check-in biométrico ofrece una solución eficiente, segura y fácil de usar para las empresas que buscan mejorar sus procesos de control de acceso y aumentar la seguridad de sus instalaciones.

**Palabras clave:** Check-in, tecnología biométrica, autenticación, verificación de identidad, empleados, entrada, salida, registro, acceso, Raspberry Pi, sensor biométrico, huellas dactilares, API REST, NodeJS, seguridad, control de acceso.



# Abstract

This project aims to implement a check-in system that uses biometric technology to quickly and securely verify the identity of employees in a company. The implementation of this system will improve efficiency in the process of employee entry and exit from the office, reducing the time and effort required for manual access control.

The system utilizes a Raspberry Pi as the hardware platform and a fingerprint biometric sensor to capture the identity of employees. The captured biometric information is processed by a NodeJS-based REST API that verifies the identity of employees and records their entry and exit from the office.

This biometric check-in system project offers an efficient, secure, and user-friendly solution for companies seeking to improve their access control processes and enhance the security of their facilities.



# Parte I

## Memoria del Proyecto



# Capítulo 1

## Introducción

### 1.1. Estado del arte

Para la realización de este proyecto se ha realizado una investigación previa en términos de análisis del mercado, disponibilidad de productos y servicios similares, y los precios de los mismos.

En un principio puede parecer que la solución propuesta es algo ya existente, lo que es correcto pero solo parcialmente, esto quiere decir que existen dispositivos que permiten la autenticación a través de características biométricas, al igual que sistemas de logIn/fichaje para los empleados de una empresa. La ventaja competitiva de esta solución consiste en la integración de ambas, evitando también el coste de compra del dispositivo de autenticación. Esta información ha sido certificada mediante la investigación. En concreto se destaca el caso de la empresa 2N, líder en dispositivos de este tipo.

Tras un contacto inicial con un comercial de la empresa, la conclusión obtenida es la siguiente:

Proveen productos de gran calidad en términos de autenticación entre otros, pero no alcanzan el nivel de integración deseado por la empresa que solicita la solución desarrollada en este proyecto. Esto quiere decir que como máximo facilitan información sobre las API's con las que se podría integrar su producto, pero no el servicio de integración en sí mismo.

Por último conviene añadir un último punto en términos del contexto legal:

La legislación reciente en España requiere que las empresas mantengan registros precisos de las horas trabajadas por sus empleados, incluyendo las horas extras. El objetivo de esta legislación es prevenir los abusos laborales y garantizar que los empleados sean compensados de manera justa por su trabajo.

La ley, conocida como Real Decreto-Ley 8/2019, se aprobó en marzo de 2019 y entró en vigor en mayo del mismo año. Requiere que las empresas lleven un registro de las horas de inicio y finalización de la jornada laboral de cada empleado, así como de cualquier descanso tomado durante el día. Teniendo esto en cuenta, la solución propuesta cobra todavía más sentido, a la hora de mejorar/facilitar/automatizar este tedioso proceso.

## 1.2. Descripción de la solución

La propuesta es un sistema de registro biométrico que emplea la tecnología de huellas dactilares para autenticar a los trabajadores de una compañía. El sistema se compone de un sensor biométrico de huellas dactilares y una Raspberry Pi que actúa como plataforma de hardware. La información biométrica obtenida por el sensor se procesa mediante una API REST basada en NodeJS que verifica la identidad del trabajador y registra su ingreso y salida de la oficina.

El sistema se integra fácilmente con el software de recursos humanos de la empresa, lo que permite una gestión sencilla de los datos de los empleados y un seguimiento del tiempo trabajado. Además, el sistema ofrece la opción de generar informes y estadísticas de asistencia, lo que facilita la comprensión de la asistencia y gestión del tiempo. Es necesario destacar que la parte de integración con dicho software de recursos humanos no comprende este proyecto, y es un desarrollo posterior y/o auxiliar.

La implementación del sistema de registro biométrico tiene varias ventajas. En primer lugar, el proceso de verificación de huellas dactilares es más rápido y preciso que otros métodos de autenticación, como el uso de tarjetas o códigos de acceso. En segundo lugar, el sistema elimina la necesidad de un control manual de acceso, lo que ahorra tiempo y

recursos para la empresa. Además, la verificación biométrica de huellas dactilares es más segura y exacta que otros métodos de autenticación.

En resumen, la solución de registro biométrico propuesta ofrece una forma fácil, segura y eficiente de autenticar a los trabajadores y controlar el acceso a las instalaciones de la empresa. La solución proporciona una gestión centralizada de la información biométrica de los empleados y el tiempo trabajado, mejorando la eficiencia y seguridad de la empresa.

### 1.3. Objetivos de la solución

Los objetivos de la solución del proyecto son los siguientes:

- Mejorar la eficiencia y la precisión del proceso de control de acceso: La implementación de la tecnología biométrica de huellas dactilares permite una verificación de la identidad más rápida, precisa y segura.
- Simplificar el proceso de control de acceso: La eliminación de un control manual de acceso ahorra tiempo y recursos, y reduce la posibilidad de errores humanos.
- Proporcionar una administración centralizada de la información biométrica de los empleados y la gestión del tiempo: El sistema se integrará fácilmente con el sistema de gestión de recursos humanos de la empresa, lo que permitirá una fácil administración de los datos de los empleados y un seguimiento del tiempo trabajado.
- Mejorar la seguridad de las instalaciones de la empresa: La tecnología biométrica de huellas dactilares es mucho más segura y precisa que otros métodos de autenticación.
- Proporcionar informes y estadísticas de la asistencia de los empleados: La solución permite generar informes y estadísticas de la asistencia de los empleados, lo que permite a la empresa tener una mejor comprensión de la asistencia de los empleados y la gestión del tiempo y la productividad.

En general, la solución de sistema de check-in biométrico tiene como objetivo mejorar la eficiencia, la seguridad y la gestión del tiempo de la empresa al simplificar el proceso

de control de acceso y proporcionar una administración centralizada de la información biométrica y la gestión del tiempo en la empresa.

## 1.4. Objetivos académicos del TFG

Los objetivos académicos del proyecto son los siguientes:

- Aprender de una manera más profesional y práctica una o varias tecnologías vistas de forma teórica durante el grado.
- Desarrollar un producto desde 0 con todos los pasos necesarios en el desarrollo del software, como pueden ser el análisis, el diseño, o el desarrollo de tests.
- Desarrollar la capacidad de adaptación a tecnologías nuevas y/o desconocidas, como puede ser por ejemplo en este caso la raspberry, el sensor biométrico y las herramientas de desarrollo para la parte de servidor.
- Trabajar a un nivel más técnico y profesional, aprovechando y utilizando los recursos que puede ofrecer la empresa para la que se desarrolla el proyecto.

## 1.5. Objetivos SMART

*Los objetivos SMART son específicos, medibles, alcanzables, relevantes y temporales. Son metas concretas que permiten analizar el desempeño de nuestros esfuerzos, ya sea en marketing o en cualquier área de una empresa que requiera ordenar y medir su trabajo de manera sistemática. (Titular.com, 2018) [7]*

Los objetivos concuerdan con la definición **SMART** (**S**pecific, **M**easurable, **A**ttainable, **R**elevant, **T**ime-bound) por las siguientes razones:

- Específicos (Specific): Cada objetivo se refiere a una funcionalidad concreta que debe ser implementada en la solución.

- **Medibles (Measurable):** Cada objetivo es medible, ya que se puede comprobar si se ha cumplido o no. Por ejemplo, se puede verificar si el dispositivo de identificación biométrica es capaz de capturar la información y enviarla al servidor.
- **Alcanzables (Attainable):** Cada objetivo es realista y alcanzable dentro del alcance del proyecto.
- **Relevantes (Relevant):** Cada objetivo es relevante para la solución, que es proporcionar una solución de identificación biométrica segura y eficiente.
- **Temporales (Time-bound):** Cada objetivo tiene un plazo de tiempo asociado para su implementación dentro del proyecto.



# Capítulo 2

## Gestión de proyecto del TFG

### 2.1. Resumen

Este proyecto se trata de un TFG de empresa, esto quiere decir que se ha llevado a cabo una colaboración con la empresa en la que trabaja el alumno con el fin de obtener un resultado que aporte beneficio a ambas partes. Es por ello que la gestión del mismo es compartida por el alumno, así como por los dos tutores, académico y de empresa.

Como se detalla más adelante, se ha desarrollado utilizando una metodología Ágil, para la que se ha utilizado como herramienta un tablero Kanban. Este tablero es útil a la hora de realizar la planificación y monitorización, explicadas en sus correspondientes apartados.

### 2.2. Metodologías

Para llevar a cabo este proyecto se ha utilizado una metodología de tipo ágil. Esta metodología se ha llevado a cabo mediante el marco de trabajo "Scrum of one".

El Scrum of One es un enfoque adaptado de Scrum, una metodología ágil de gestión de proyectos, diseñado específicamente para individuos que trabajan en proyectos personales o que desean aplicar principios ágiles a su trabajo individual.

En Scrum of One, una persona se desempeña como "equipo" y sigue los roles, eventos y

artefactos básicos de Scrum, pero adaptados a una escala individual. El objetivo principal es mejorar la productividad, la organización y la visibilidad del trabajo individual, al tiempo que se fomenta la adaptabilidad y la mejora continua.

Algunos de los elementos clave del Scrum of One incluyen:

**Backlog personal:** Es una lista priorizada de tareas y actividades que se deben realizar. Puede incluir elementos más grandes o pequeños, y se actualiza continuamente a medida que se completan tareas o se identifican nuevas.

**Sprint personal:** Se establecen intervalos de tiempo fijos, conocidos como "sprints", durante los cuales se enfoca en trabajar en las tareas prioritarias del backlog personal. Al final de cada sprint, se revisa el progreso y se realiza una retrospectiva para mejorar el rendimiento.

**Revisiones y retrospectivas:** Al finalizar cada sprint, se realiza una revisión para evaluar el trabajo realizado y hacer ajustes en el backlog personal. Además, se lleva a cabo una retrospectiva para reflexionar sobre el sprint y identificar mejoras para futuros sprints.

Scrum of One proporciona un marco estructurado que ayuda a las personas a organizar su trabajo, establecer metas claras, adaptarse a los cambios y mantener un enfoque constante en la entrega de valor. Aunque es una adaptación de Scrum para el trabajo individual, sigue los principios fundamentales de transparencia, inspección y adaptación de Scrum.

### 2.3. Colaboración con tutores

Para llevar a cabo el proyecto, se ha llevado a cabo una colaboración continua con los dos tutores asignados.

Por un lado, se ha trabajado con el tutor de Brooktec para desarrollar la parte técnica. Su colaboración ha sido esencial a la hora de aclarar dudas de concepto y sobre todo conseguir una calidad técnica a la altura de los requisitos.

Las reuniones se han llevado a cabo mediante la plataforma Google Meet sin seguir un orden marcado, es decir, en horario de trabajo, se ha recurrido al tutor en momentos

compatibles para las dos partes, tutor y alumno.

También se ha utilizado la plataforma Slack para comunicaciones sencillas, ya que es la que se utiliza de manera predeterminada en la empresa.

Por el otro lado, la colaboración con el tutor de la escuela ha aportado conocimiento y ayuda para la parte más relacionada con la gestión de proyectos, ya que es la especialidad de dicho tutor.

Se han llevado a cabo reuniones periódicas concertadas por correo electrónico y efectuadas a través de la plataforma Zoom. A partir del mes de marzo se han llevado a término al menos dos reuniones al mes, ya sea para resolución de dudas o simple seguimiento/monitorización del progreso.

## 2.4. Planificación

Para la planificación se ha optado por la utilización de un tablero kanban, en concreto utilizando la plataforma gratuita “Trello”.

Siguiendo la estructura del proyecto se han establecido las tareas con sus correspondientes subtareas necesarias para conseguir el objetivo propuesto.

En este tablero se han utilizado las columnas:

- Lista de tareas
- En proceso
- Hecho
- Validación con tutor

Estas columnas tienen como objetivo representar los diferentes estados de la tarea. Por ejemplo, cuando el alumno considera que ha terminado una tarea, la pondrá en la columna “Validación con tutor” para la validación de esta. Una vez validada, el tutor podrá colocarla en la columna de tareas terminadas, que tiene por nombre “hecho”.

Se ha considerado este método por las siguientes razones:

- Organización: Tener una vista global del estado del proyecto ayuda a gestionar mejor el tiempo y las tareas pendientes.
- Comunicación: Gracias a esta plataforma, que es online, se podrá colaborar con el tutor para el desarrollo de las tareas, y este podrá ver en todo momento el progreso del alumno.
- Eficiencia: la utilización de una herramienta como esta tiene como resultado un ahorro de tiempo considerable a la hora de comprobar el progreso realizado, evitando comprobaciones innecesarias de secciones ya resueltas.

## 2.5. Recursos

A continuación se detallan los recursos utilizados para llevar a cabo el proyecto, se dividen en dos secciones, recursos materiales y humanos:

### 2.5.1. Recursos Materiales

*Los recursos materiales son todos los elementos, materias primas, herramientas, máquinas, equipos y todo elemento físico que se requieren para realizar el proceso de producción de una empresa. (Quiroa, 2020) [14]*

Para este proyecto se consideran los siguientes:

- Sensor biométrico: Se encargará de recoger la información biométrica del usuario, en este caso se trata de un sensor de huellas.
- Raspberry PI: Será el dispositivo encargado de procesar la citada información del usuario, y establecer la conexión pertinente con el servidor para las diferentes operaciones requeridas.
- Ordenador portátil: Se utilizará tanto para la programación del software necesario como para la elaboración del presente documento.

- Software: Herramienta para elaboración de diagramas y paquete ofimático para documento y presentación, entre otros.
- Recursos básicos, como lo pueden ser la luz o el acceso a internet

### 2.5.2. Recursos Humanos

*Los recursos humanos (RRHH) son el conjunto de trabajadores, así como cualquier persona física que se encuentran dentro (o vinculado directamente) de una organización, sector o economía. También, en la administración de empresas, hace referencia a la gestión que se hace de los trabajadores dentro de una organización determinada. (Coll,2020) [11]*

En este caso se considerarán las horas de trabajo del alumno, a la hora de desarrollar el proyecto, y también por otro lado las horas de trabajo de los tutores en conjunto con el alumno con respecto a las reuniones programadas de ayuda técnica y monitorización.

## 2.6. Presupuesto

Como en todo proyecto, se ha realizado una planificación del presupuesto que supondrá el proyecto. El presupuesto es el costo total necesario para completar el proyecto en un determinado periodo de tiempo y con la finalidad de obtener unos resultados específicos.

Se detalla el presupuesto en dos secciones: las tareas y los recursos necesarios para realizarlas.

Se puede observar en la tabla el calendario y el presupuesto planificado para este proyecto. Como se puede apreciar muchas de la tareas comienzan y finalizan en fechas muy próximas, esto se debe a que el trabajo se realiza por capas, haciendo en estas etapas cada una de las fases de análisis, diseño e implementación.

Además, se puede apreciar que tanto las horas estimadas con las horas realizadas finalmente exceden el total de horas, 300 horas, que representan los 12 créditos que representa

## Capítulo 2. Gestión de proyecto del TFG

Tarea	Fecha de inicio	Fecha de finalización	Participantes	Horas estimadas	Horas realizadas	Coste
Matriculación curso 2022-2023	29/06/2022	29/06/2022	Jorge	0	0	
Brainstorming de ideas para TFG	22/01/2023	29/01/2023	Jorge	6	6	87,72 €
Reunión de inicio de proyecto	14/02/2023	14/02/2023	Jorge, Tutor	2	2	29,24 €
Trabajo de estudio del tema del proyecto	22/01/2023	10/02/2023	Jorge	10	12	175,44 €
Realización de la introducción y estado del arte	14/02/2023	24/02/2023	Jorge	20	24	350,88 €
Análisis y especificación de requisitos	24/02/2023	04/03/2023	Jorge	18	15	219,30 €
Aprendizaje de herramientas y lenguaje de programación	22/01/2023	20/04/2023	Jorge	25	30	438,60 €
Análisis	04/03/2023	28/06/2023	Jorge	20	16	233,92 €
Diseño	11/03/2023	28/06/2023	Jorge	20	25	365,50 €
Implementación del proyecto	15/03/2023	02/07/2023	Jorge	100	105	1.535,10 €
Pruebas	10/06/2023	02/07/2023	Jorge	15	15	219,30 €
Elaboración Memoria	14/02/2023	08/07/2023	Jorge	50	50	731,00 €
Poblar Base de datos	25/03/2023	05/04/2023	Jorge	5	10	146,20 €
Reuniones iniciales de seguimiento	25/02/2023	06/04/2023	Jorge, Tutor	16	16	233,92 €
Reuniones finales de seguimiento	06/04/2023	11/07/2023	Jorge, Tutor	20	25	365,50 €
Elaboración de la presentación	28/06/2023	11/07/2023	Jorge	8	8	116,96 €
			Total:	335	359	5.248,58 €

Figura 2.1: Calendario del proyecto

Recursos a utilizar	Fecha inicio	Fecha fin	Descripción	Precio inicial	Precio total
Ordenador personal	14/02/2023	11/07/2023	Ordenador portátil personal del alumno	1000	1000
Raspberry PI	14/02/2023	11/07/2023	Dispositivo para reconocimiento de huellas	229,99	229,99
Sensor biométrico	14/02/2023	11/07/2023	Dispositivo para reconocimiento de huellas	19,99	19,99
Microsoft Office Estudiantes 2019	14/02/2023	11/07/2023	Software para realizar la memoria	149	149
Visual Paradigm	14/02/2022	11/07/2023	Software para realizar diagramas del proyecto	99	99
Luz	14/02/2022	11/07/2023	Premio medio de luz en España por 3,45 KW	35,5	177,5
Internet	14/02/2022	11/07/2023	Red para transmitir y obtener información	50	250
			Total:		1925,48

Figura 2.2: Estimación y monitorización de los recursos

el trabajo de fin de grado.

Asimismo, se explicará si he ha usado alguna metodología especial (metodologías ágiles, TDD, etc).

### 2.6.1. Estimación y monitorización de los recursos a utilizar

Aunque los recursos han sido proporcionados por una parte por la Universidad de Valladolid y por otra parte por el alumno, la tabla 2.2 representa los recursos y los costes de mercado utilizados en el marco del proyecto.

## 2.7. DAFO - Debilidades Amenazas Fortalezas y Oportunidades

Totales	Unidades/Horas	Precio unitario	Precio Total
Trabajo del proyectante	339	14,62	4956,18
Recursos	1	0	1925,48
		Total:	7215,08

Figura 2.3: Presupuesto final del proyecto

### 2.6.2. Presupuesto total

En la tabla 2.3 se representa el presupuesto total que supone el proyecto. Este valor procede de la suma de los recursos que han sido necesarios más el trabajo realizado por el estudiante. Para valorar el trabajo en horas del alumno se ha establecido un salario medio cobrado por ingeniero informático en España, 14,62 €/ hora.

## 2.7. DAFO - Debilidades Amenazas Fortalezas y Oportunidades

*El análisis DAFO (o FODA) es una herramienta de gestión que facilita el proceso de planeación estratégica, proporcionando la información necesaria para la implementación de acciones y medidas correctivas (Díaz y Matamoros, 2006 ) [13]*

Tras analizar las condiciones concretas del proyecto, se presenta el DAFO correspondiente en la figura 2.4

DAFO	Negativo	Positivo
	<b>Debilidades</b>	<b>Fortalezas</b>
Interno	- Tutor y estudiante tienen otros compromisos laborales	- Es un TFG de empresa
	- El software del equipo está expuesto a fallos	
	- El estudiante no ha terminado los créditos previos al TFG	
	<b>Amenazas</b>	<b>Oportunidades</b>
Externo	- Enfermar	- La empresa puede brindar conocimientos e infraestructura al proyecto
	- Llegada de la fecha de presentación que exige tener créditos terminados	- Acceso a expertos y profesionales de la empresa: Se puede aprovechar la experiencia y conocimientos de los expertos y profesionales de la empresa para obtener orientación y asesoramiento específico

Figura 2.4: Presupuesto final del proyecto

## 2.8. Gestión de riesgos

Tras un análisis de los requisitos, especificaciones y condiciones del proyecto se han detectado los siguientes riesgos para la ejecución del mismo:

### 2.8.1. RSG01 Ausencia temporal del estudiante por enfermedad

<u>Descripción:</u>	El estudiante no puede dedicar tiempo a trabajar en el TFG durante un período determinado debido a causas médicas que le impidan trabajar con normalidad.
<u>Probabilidad:</u>	media
<u>Impacto:</u>	bajo
<u>Acciones de mitigación:</u>	<ul style="list-style-type: none"> <li>▪ Se planifica con un colchón temporal que contemple posibles contingencias.</li> <li>▪ Si no afecta al camino crítico, se reordenarán las tareas.</li> </ul>
<u>Acciones Correctivas:</u>	<ul style="list-style-type: none"> <li>▪ Recuperar el tiempo perdido distribuyéndolo entre los siguientes días.</li> <li>▪ Si la duración es más prolongada, pasar tareas al siguiente sprint.</li> <li>▪ Si ocurre en los últimos sprints y las dos anteriores no se pueden realizar, hacer uso de sprints de refuerzo.</li> </ul>

## 2.8.2. RSG02 Fallos técnicos en el equipo de trabajo del estudiante

<u>Descripción:</u>	El equipo de trabajo del estudiante sufre algún fallo técnico y no se puede continuar trabajando con normalidad.
<u>Probabilidad:</u>	baja
<u>Impacto:</u>	alto
<u>Acciones de mitigación:</u>	<ul style="list-style-type: none"> <li>▪ Hacer una copia de seguridad de los documentos importantes cada poco tiempo.</li> <li>▪ Mantener una copia de los documentos importantes en algún repositorio online.</li> <li>▪ Trabajar en lo posible de forma online con los documentos (Overleaf, Microsoft Teams...).</li> </ul>
<u>Acciones Correctivas:</u>	<ul style="list-style-type: none"> <li>▪ Si se ha realizado copias de seguridad de los documentos, recuperarlo desde otro dispositivo para seguir trabajando.</li> <li>▪ Si no se ha realizado copias de seguridad, tratar de recuperar los documentos con los mecanismos de recuperación que ofrecen los propios programas.</li> </ul>
<u>Estado del riesgo:</u>	abierto

### 2.8.3. RSG03 Mala estimación de tiempo y trabajo de algunas actividades

<u>Descripción:</u>	La estimación producida para algunas actividades o tareas a realizar en los sprints resulta ser muy sesgada y el tiempo empleado realmente no se asemeje a lo estimado, habiendo desviación no contempladas.
<u>Probabilidad:</u>	media
<u>Impacto:</u>	medio
<u>Acciones de mitigación:</u>	<ul style="list-style-type: none"> <li>▪ Establecer uno o más sprints de refuerzo al final del proceso de desarrollo del proyecto que puedan cubrir estas necesidades de tiempo adicional.</li> <li>▪ Ser realista en las estimaciones. Para ello, mejorar con la práctica.</li> <li>▪ Dejar los detalles pequeños para el final y centrarse más en las partes importantes del proyecto.</li> </ul>
<u>Acciones Correctivas:</u>	<ul style="list-style-type: none"> <li>▪ Hacer uso de los Sprints de refuerzo.</li> <li>▪ Si no hay disponibilidad de más sprints de refuerzo, recortar el alcance del proyecto en la medida de lo posible.</li> </ul>
<u>Estado del riesgo:</u>	abierto

#### 2.8.4. RSG04 Ausencia temporal del tutor por causas no planificadas

<u>Descripción:</u>	El tutor de empresa o académico no puede continuar con el seguimiento del TFG durante un período determinado debido a causas no planificadas que le impidan trabajar con normalidad.
<u>Probabilidad:</u>	baja
<u>Impacto:</u>	bajo
<u>Acciones de mitigación:</u>	<ul style="list-style-type: none"> <li>■ Tener contacto asiduo para poder saber si el riesgo se materializa en algún momento.</li> <li>■ Reservar fechas alternativas para las reuniones en caso de posibles problemas.</li> </ul>
<u>Acciones Correctivas:</u>	<ul style="list-style-type: none"> <li>■ Seguir trabajando en el TFG hasta poder tener la siguiente reunión con el tutor, e ir apuntando dudas o problemas que surjan.</li> <li>■ Si es posible comunicarse mediante otros medios que no sean videoconferencia (chats, emails...), enviar dudas o un resumen del contenido que se vería en la reunión.</li> </ul>
<u>Estado del riesgo:</u>	abierto

# Capítulo 3

## Requisitos de la solución

### 3.1. Requisitos Funcionales

#### 3.1.1. RFA01 Biometría

**Descripción:** El dispositivo de identificación biométrica debe ser capaz de capturar la información biométrica del usuario.

**Entradas:** Acceso al dispositivo de identificación.

**Proceso:**

- Se coloca el dedo en el sensor
- El láser (sistema) lee la huella
- Codifica la huella (en información digital)
- Guarda la información.

**Salidas:** Se ha recogido la información biométrica del usuario.

**Restricciones:** El dispositivo debe estar configurado con la librería asociada al sensor para funcionar.

**Requisitos relacionados:** RFA02

### 3.1.2. RFA02 Autenticación

**Descripción:** El servidor debe ser capaz de autenticar y autorizar el acceso del usuario.

**Entradas:** Huella ya registrada. Huella codificada, nombre de usuario y token de autenticación.

**Proceso:**

- Recibe la información
- Decodifica el token
- Comprueba validez del token
- Comprueba tipo de dispositivo emisor
- Autentica y autoriza el acceso.

**Salidas:** Usuario autenticado.

**Restricciones:** El dispositivo debe indicar si es 'deviceó 'client' mediante el token.

**Requisitos relacionados:** RFA01

### 3.1.3. RFA03 API Rest

**Descripción:** El servidor debe ser capaz de proporcionar una API REST para las funcionalidades necesarias, como registro de usuario, inicio de sesión, consulta de infor-

mación, etc.

#### 3.1.4. RFA04 Comunicación dispositivo servidor

**Descripción:** La comunicación entre el dispositivo y el servidor debe ser segura y autenticada.

**Entradas:** El usuario ha introducido su huella en el dispositivo de autenticación.

**Proceso:**

- Dispositivo envía información junto con token de autenticación
- Servidor recibe la información
- Servidor decodifica token y verifica la autenticidad del dispositivo.

**Salidas:** Dispositivo autenticado de forma segura.

**Restricciones:** Debe existir una conexión estable entre dispositivo y servidor.

**Requisitos relacionados:** RFA01, RFA02

## 3.2. Requisitos No Funcionales

### 3.2.1. RNFA01 Raspberry

**Descripción:** El dispositivo de identificación biométrica debe ser compatible con Raspberry Pi suministrado por Brooktec.

### 3.2.2. RNFA02 Tecnologías Servidor

**Descripción:** El servidor debe estar desarrollado en Node.js y utilizar Express como marco de trabajo.

### **3.2.3. RNFA03 Persistencia de datos**

**Descripción:** La persistencia de datos puede ser definida libremente.

### **3.2.4. RNFA04 JWT**

**Descripción:** Se debe utilizar JWT para autenticación.

### **3.2.5. RNFA05 UUID**

**Descripción:** Se debe utilizar UUID para identificación de entidades.

### **3.2.6. RNFA06 Estándares de calidad**

**Descripción:** El código del servidor debe cumplir con los estándares de calidad establecidos.

### **3.2.7. RNFA07 Linter**

**Descripción:** Brooktec Eslint debe ser utilizado como herramienta de calidad de código.

### **3.2.8. RNFA08 Cobertura de pruebas**

**Descripción:** La cobertura de pruebas debe ser del 90

### **3.2.9. RNFA09 Repositorio**

**Descripción:** El código debe ser alojado en un repositorio de la compañía, utilizando el flujo de trabajo estándar gitflow.

### 3.2.10. RNFA01 Raspberry

**Descripción:** El dispositivo de identificación biométrica debe ser compatible con Raspberry Pi suministrado por Brooktec.

### 3.2.11. RNFA02 Tecnologías Servidor

**Descripción:** El servidor debe estar desarrollado en Node.js y utilizar Express como marco de trabajo.

### 3.2.12. RNFA03 Persistencia de datos

**Descripción:** La persistencia de datos puede ser definida libremente.

### 3.2.13. RNFA04 JWT

**Descripción:** Se debe utilizar JWT para autenticación.

### 3.2.14. RNFA05 UUID

**Descripción:** Se debe utilizar UUID para identificación de entidades.

### 3.2.15. RNFA06 Estándares de calidad

**Descripción:** El código del servidor debe cumplir con los estándares de calidad establecidos.

### 3.2.16. RNFA07 Linter

**Descripción:** Brooktec Eslint debe ser utilizado como herramienta de calidad de código.

### 3.2.17. RNFA08 Cobertura de pruebas

**Descripción:** La cobertura de pruebas debe ser del 90

### 3.2.18. RNFA09 Repositorio

**Descripción:** El código debe ser alojado en un repositorio de la compañía, utilizando el flujo de trabajo estándar gitflow.

# Capítulo 4

## Modelo de dominio

El modelo de dominio es una representación conceptual de los elementos clave y las relaciones dentro de una determinada área de conocimiento. Es una herramienta utilizada en el análisis de sistemas de software para comprender y documentar la estructura y el comportamiento de un dominio específico.

El objetivo principal del modelo de dominio es capturar y representar los conceptos fundamentales y las relaciones entre ellos en un dominio de negocio o aplicación.

En la figura 4.1 se puede observar el modelo de dominio correspondiente a este desarrollo:

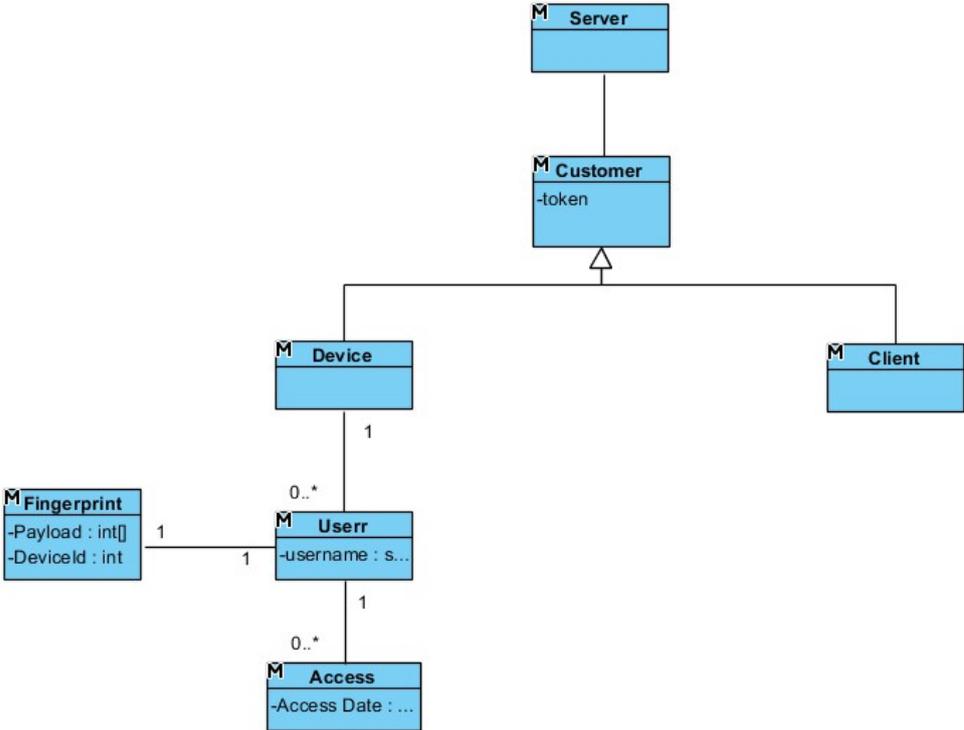


Figura 4.1: Modelo de dominio

# Capítulo 5

## Análisis

### 5.1. Diagrama de casos de uso

*Un caso de uso representa una interacción típica entre un usuario y un sistema informático. Utilizaremos los casos de uso para capturar los requisitos funcionales del sistema (Microsoft Corporation, 2015) [10]*

En el diagrama de la figura 5.1 se representan los principales casos de uso del sistema, además de los principales actores que intervienen en ellos, posteriormente se definirán y explicarán por separado:

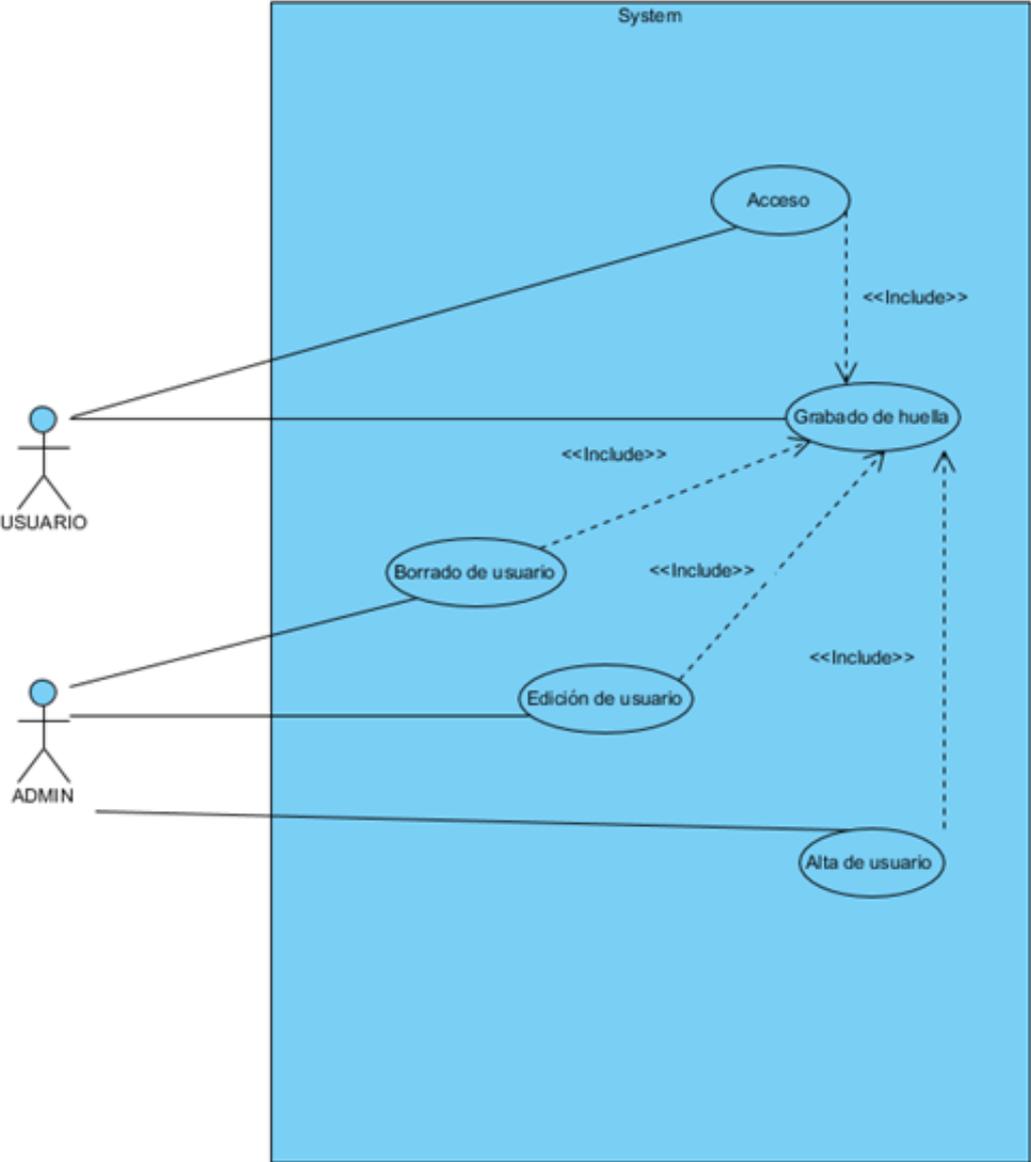


Figura 5.1: Diagrama de casos de uso

## 5.2. Casos de uso

### 5.2.1. CU01 Grabado de huella

<b>Nombre e ID del CU</b>	CU-01. Grabado de huella
<b>Actor</b>	ADMIN, USUARIO
<b>Descripción</b>	El usuario envía sus datos de usuario al sistema, generando una solicitud de alta que deberá ser gestionada por un administrador.
<b>Precondiciones</b>	Conexión a internet, dispositivo en funcionamiento.
<b>Postcondiciones</b>	Huella grabada tanto en el servidor como localmente en el dispositivo (rpi)
<b>Escenario principal</b>	<p>FN1 ADMIN selecciona el programa de acceso</p> <p>FN2 Sistema solicita al usuario la introducción del nombre de usuario</p> <p>FN3 USUARIO introduce el nombre de usuario</p> <p>FN4 SISTEMA solicita al usuario la introducción del id interno del usuario en el sensor</p> <p>FN5 USUARIO introduce el id solicitado</p> <p>FN6 SISTEMA solicita al usuario una primera introducción de la huella</p> <p>FN7 USUARIO introduce la huella</p> <p>FN8 SISTEMA solicita al usuario una segunda introducción de la huella</p> <p>FN9 USUARIO introduce la huella</p> <p>FN10 SISTEMA crea el modelo de la huella y almacena tanto en el dispositivo como en el servidor la huella y los datos del usuario.</p>

Cuadro 5.1: CU-01. Grabado de huella

### 5.2.2. CU02 Alta de usuario

<b>Nombre e ID del CU</b>	CU-02. Alta de usuario
<b>Actor</b>	ADMIN
<b>Descripción</b>	ADMIN introduce el host, el nuevo nombre de usuario y la huella codificada en una petición POST al sistema. El sistema confirma la corrección de los datos y añade un nuevo usuario a la base de datos.
<b>Precondiciones</b>	Conexión a internet, CU01 Grabado de huella
<b>Postcondiciones</b>	Usuario presente en la base de datos
<b>Escenario principal</b>	<p>SP1 ADMIN introduce el host, el nuevo nombre de usuario y la huella codificada en una petición POST al sistema.</p> <p>SP2 El sistema confirma la corrección de los datos y añade un nuevo usuario al sistema/DB.</p>
<b>Flujos alternativos</b>	<p>FA2a El sistema rechaza el alta por token no correcto.</p> <p>FA2b El sistema rechaza el alta por nombre de usuario ya existente.</p>

Cuadro 5.2: CU-02. Alta de usuario

## 5.2.3. CU03 Borrado de usuario

<b>Nombre e ID del CU</b>	CU-03. Borrado de usuario
<b>Actor</b>	ADMIN
<b>Descripción</b>	ADMIN introduce el host y el nombre de usuario a borrar en una petición DELETE al sistema. El sistema confirma la corrección de los datos y elimina el usuario de la base de datos.
<b>Precondiciones</b>	Conexión a internet, CU01 Grabado de huella
<b>Postcondiciones</b>	Usuario eliminado de la base de datos
<b>Escenario principal</b>	<p>SP1 ADMIN introduce el host y el nombre de usuario a borrar en una petición DELETE al sistema.</p> <p>SP2 El sistema confirma la corrección de los datos y elimina el usuario del sistema/DB.</p>
<b>Flujos alternativos</b>	<p>FA3a El sistema rechaza el borrado por token no correcto.</p> <p>FA3b El sistema rechaza el borrado por nombre de usuario no existente/ya borrado.</p>

Cuadro 5.3: CU-03. Borrado de usuario

### 5.2.4. CU04 Edición de usuario

<b>Nombre e ID del CU</b>	CU-04. Edición de usuario
<b>Actor</b>	ADMIN
<b>Descripción</b>	ADMIN introduce el host, el nombre de usuario y la huella codificada en una petición PUT al sistema. El sistema confirma la corrección de los datos y edita el usuario en la base de datos.
<b>Precondiciones</b>	
<b>Postcondiciones</b>	Usuario editado en la base de datos, CU01 Grabado de huella
<b>Escenario principal</b>	<p>SP1 ADMIN introduce el host, el nombre de usuario y la huella codificada en una petición PUT al sistema.</p> <p>SP2 El sistema confirma la corrección de los datos y edita el usuario en la base de datos.</p>
<b>Flujos alternativos</b>	<p>FA4a El sistema rechaza la edición por token no correcto.</p> <p>FA4b El sistema rechaza la edición por nombre de usuario no existente.</p>

Cuadro 5.4: CU-04. Edición de usuario

### 5.2.5. CU05 Acceso de usuario

<b>Nombre e ID del CU</b>	CU-05. Acceso de usuario
<b>Actor</b>	USUARIO
<b>Descripción</b>	El USUARIO coloca el dedo en el sensor. El SISTEMA reconoce la huella del USUARIO y registra la entrada en la base de datos, mostrando una luz verde.
<b>Precondiciones</b>	Conexión a internet, CU01 Grabado de huella
<b>Postcondiciones</b>	Registro de entrada en la base de datos
<b>Escenario principal</b>	<p>SP1 El USUARIO coloca el dedo en el sensor.</p> <p>SP2 El SISTEMA reconoce la huella del USUARIO y registra la entrada en la base de datos.</p> <p>SP3 El SISTEMA muestra una luz verde.</p>
<b>Flujos alternativos</b>	<p>FA5a El SISTEMA reconoce la huella del USUARIO pero no está en la base de datos.</p> <p>FA5b El SISTEMA no reconoce la huella del USUARIO.</p>

Cuadro 5.5: CU-05. Acceso de usuario

## 5.3. Diagrama de clases de análisis

Un diagrama de clases de análisis es una herramienta utilizada en el análisis de sistemas y el diseño orientado a objetos para representar las clases y las relaciones entre ellas en un dominio de problema específico

El objetivo principal del diagrama de clases de análisis es proporcionar una representación visual clara y concisa de la estructura estática del sistema en términos de clases

y sus relaciones. Ayudará a comprender y comunicar cómo se organizan y se relacionan los objetos en el sistema, y a identificar las responsabilidades y las interacciones entre las clases.

A continuación se muestra el diagrama de clases de análisis de este sistema:

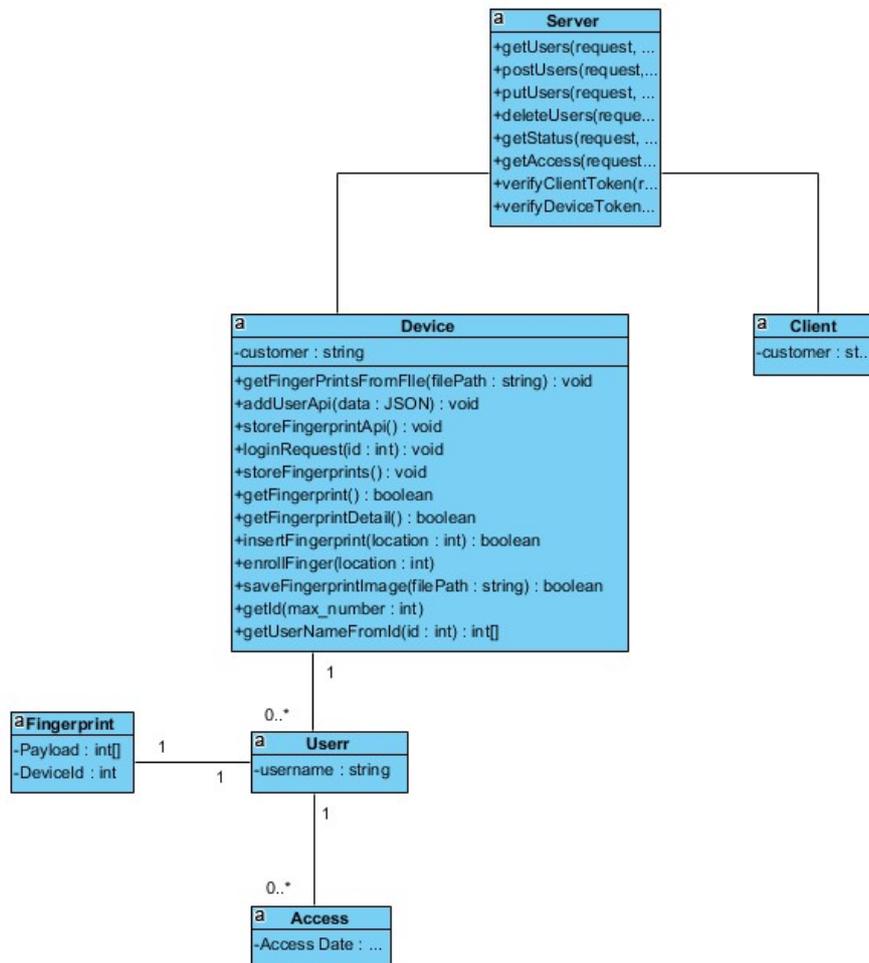


Figura 5.2: Diagrama de clases de análisis

## 5.4. Diagramas de secuencia

*Dentro de los diagramas de comportamiento en UML que permiten enfatizar las interacciones entre los objetos se encuentran los diagramas de secuencias, este describe el comportamiento del sistema y las operaciones que se realizan representando los objetos y los mensajes que se intercambian, ya que en un sistema real y funcional los objetos interactúan entre sí, y tales iteraciones suceden con el tiempo que se asigna, es decir que el diagrama de secuencias de UML es una mecánica de interacción en base a los tiempos.* (Wordpress,2015)[2]

Para este proyecto, se han modelado dos diagramas de secuencia que se corresponden con los dos casos considerados principales del conjunto, como lo son el grabado de huella (CU01) y el acceso de usuario (CU05). A continuación se muestran ambos diagramas:

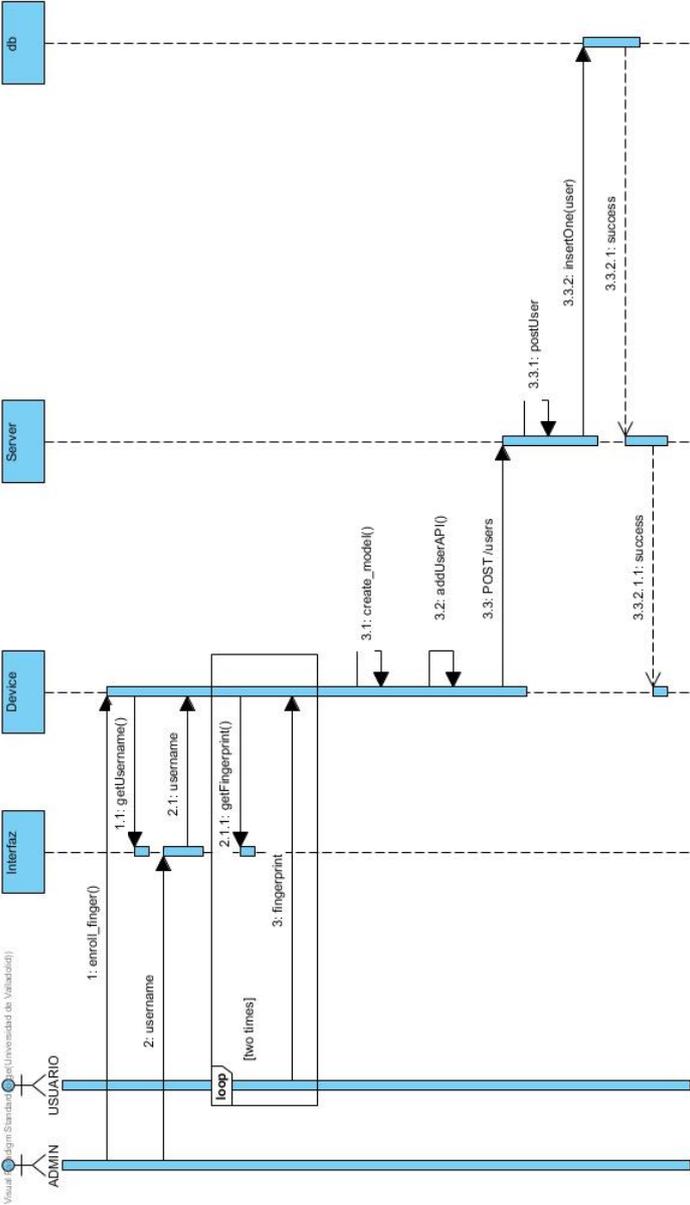


Figura 5.3: Registro de huella

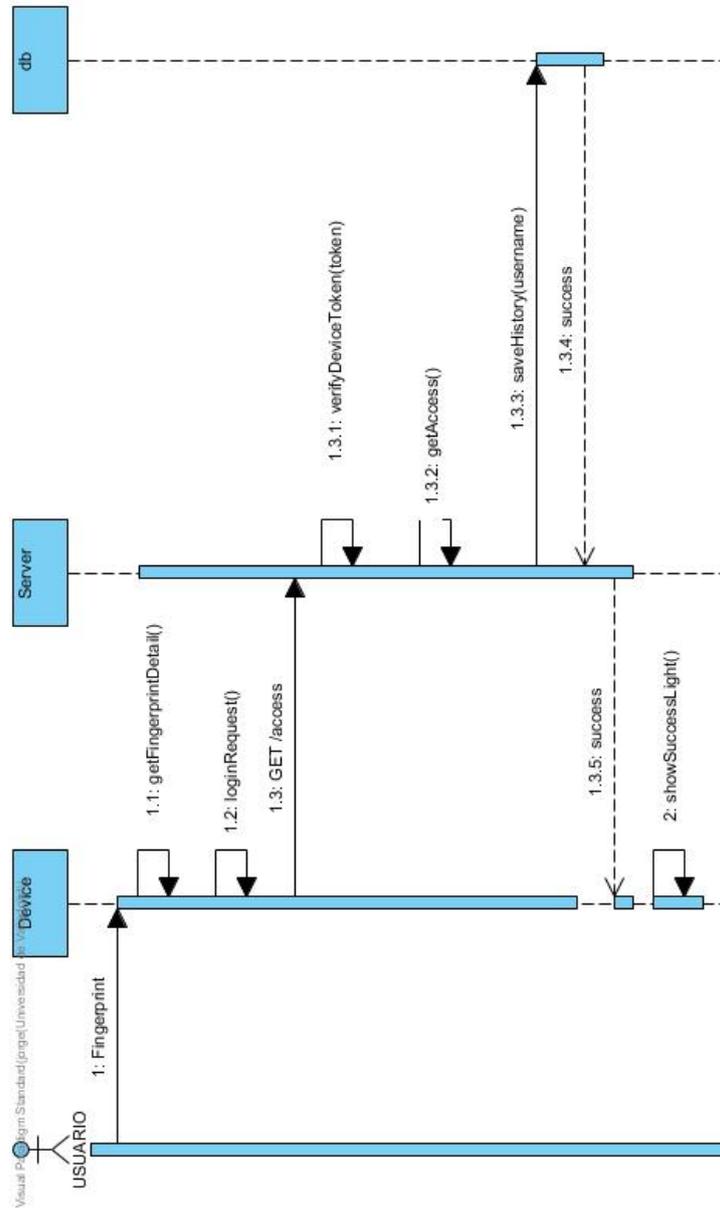


Figura 5.4: Acceso



# Capítulo 6

## Diseño

### 6.1. Diagrama de estados

*Un diagrama de estado UML (también llamado diagrama de estado, diagrama de transición de estados o diagrama de máquina de estados) muestra los estados por los que pasa una máquina de estados finitos, es decir, un modelo de comportamiento que consiste en acciones y estados o transiciones a otros estados. (Desarrollo web, 2020) [6]*

A continuación se puede ver el diagrama de estados correspondiente al registro de la huella de un usuario (CU01)

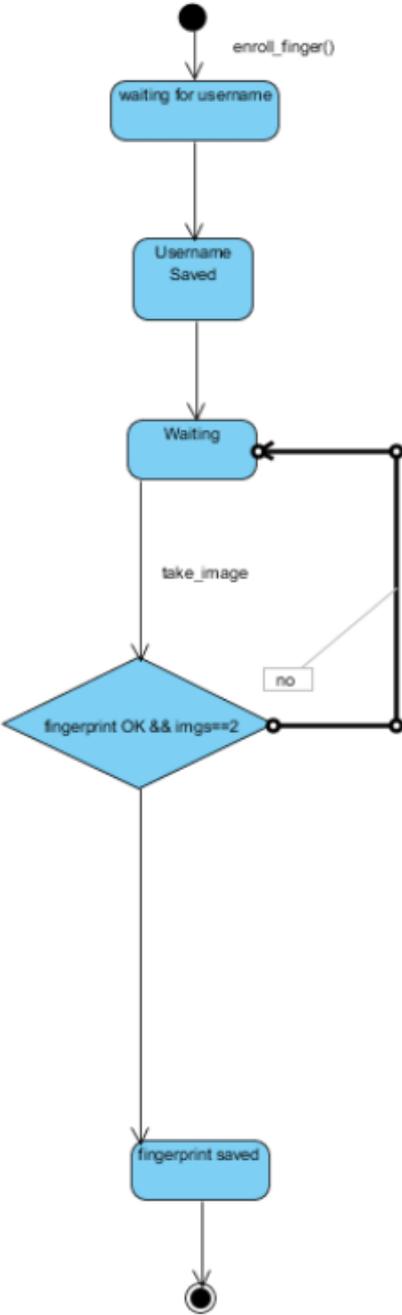


Figura 6.1: Diagrama de estados

## 6.2. Diagrama de despliegue

El diagrama de despliegue es una herramienta esencial en el análisis y diseño de sistemas de software, que permite representar la distribución física de los componentes de una aplicación y cómo interactúan entre sí en un entorno de despliegue específico.

El objetivo principal del diagrama de despliegue es visualizar la infraestructura tecnológica y los recursos necesarios para que la aplicación funcione correctamente. Esto incluye hardware, software, redes, servidores y otros componentes relevantes.

En este diagrama podremos observar tres elementos de tipo device (servidor y dos clientes) que se conectan mediante el protocolo IP. Además se indican los sistemas operativos en los que se ejecutan estos, incluyendo algunos detalles relevantes como puede ser la librería “Adafruit-Fingerprint”, esencial en este proyecto para el funcionamiento de la rapsberry con el sensor.

A continuación se presenta el diagrama de despliegue:

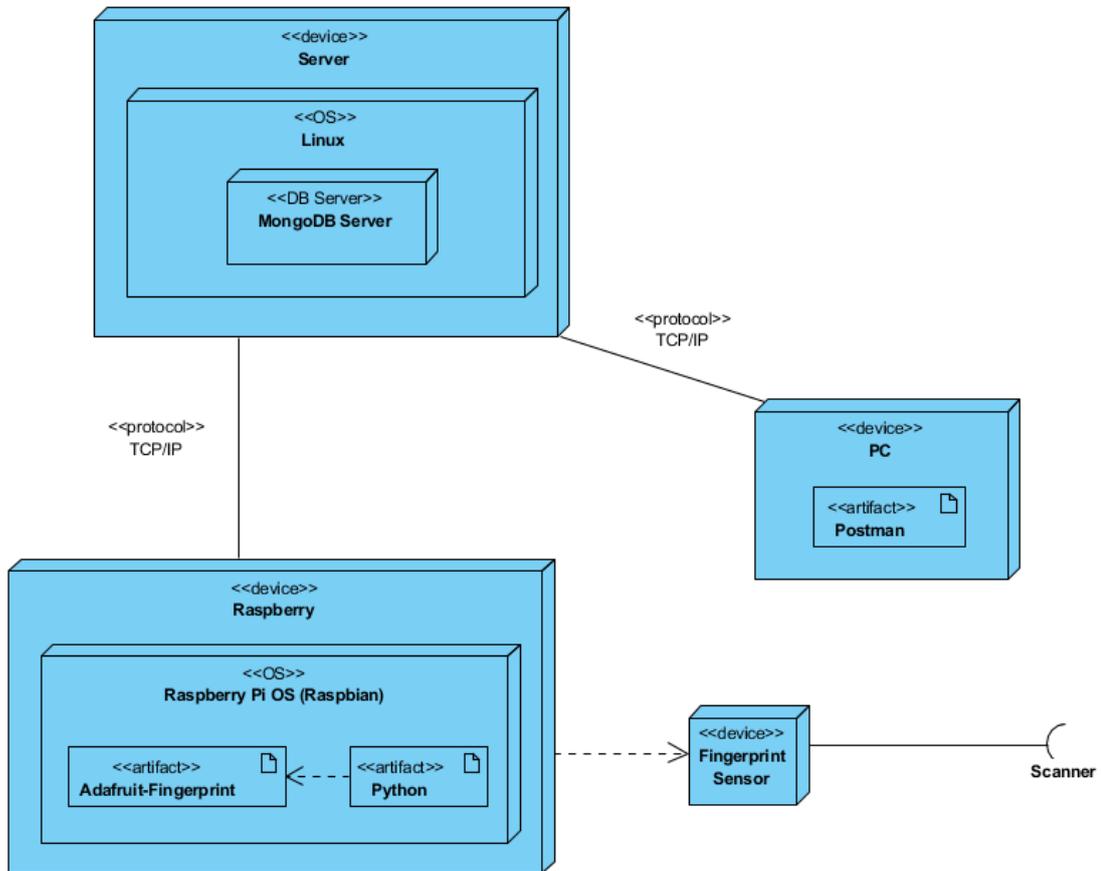


Figura 6.2: Diagrama de despliegue

### 6.3. Diagrama de clases de diseño

El diagrama de clases de diseño es una herramienta fundamental en el proceso de desarrollo de software que permite representar la estructura y las relaciones entre las clases en un sistema de manera detallada.

El objetivo principal del diagrama de clases de diseño es capturar y representar las clases, atributos, métodos y relaciones necesarios para construir el sistema de software. A diferencia del diagrama de clases de análisis, el diagrama de clases de diseño se centra en aspectos más técnicos y específicos de implementación. Proporciona una visión detallada

de cómo las clases interactúan entre sí, cómo se heredan propiedades y comportamientos, y cómo se gestionan las dependencias y las colaboraciones entre los componentes del sistema.

A continuación se presenta el diagrama de clases de diseño:

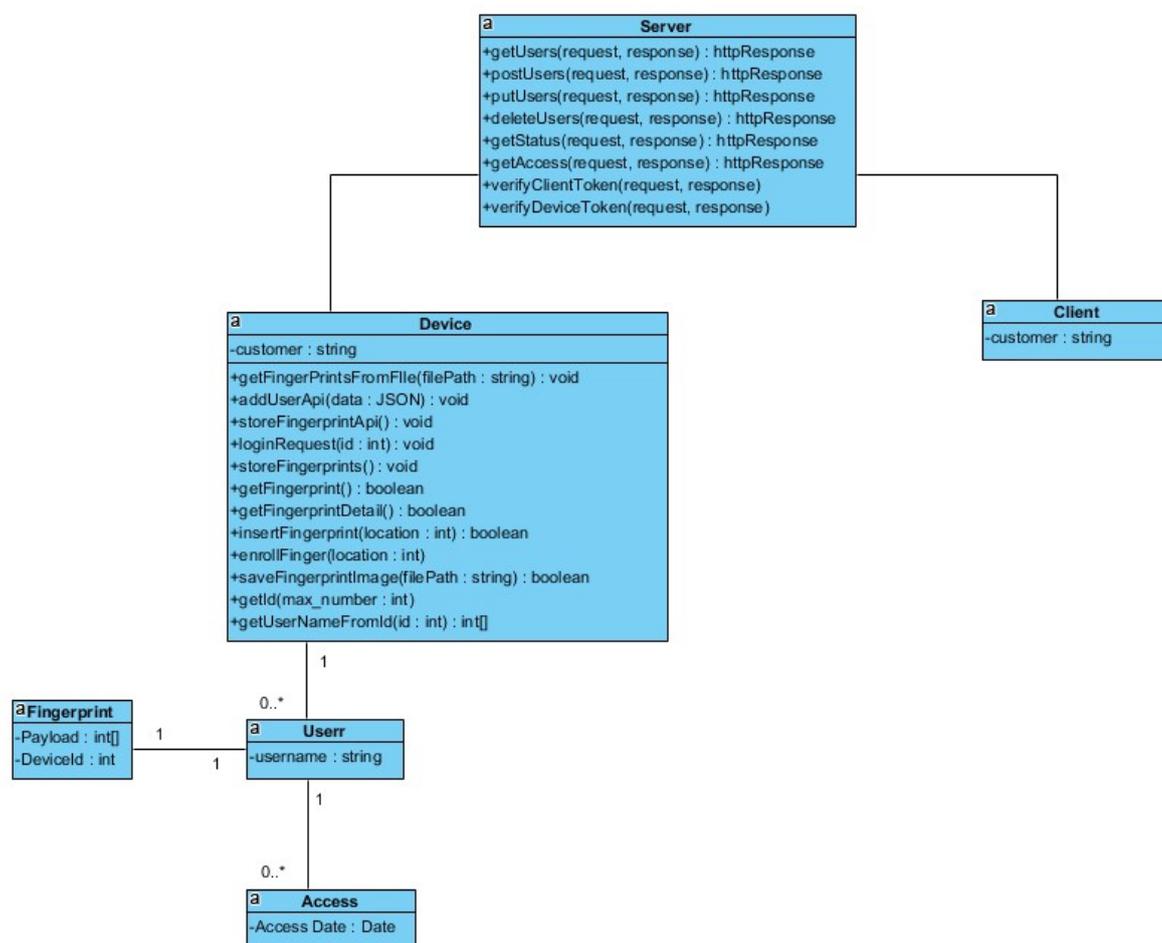


Figura 6.3: Clases de diseño

## 6.4. Diagrama de base de datos

El diagrama de base de datos es una herramienta esencial en el diseño de sistemas de software que permite representar la estructura y las relaciones de los datos en una base de datos. Aunque en este caso es un diagrama sencillo, conviene incluirlo para tener una visión global del sistema y su funcionamiento.

Como se puede ver en la figura 9, se trata de una tabla “users” que almacena información de los empleados de la empresa, que se conecta con una tabla “history”, que representa el histórico de accesos de los usuarios a la oficina.



Figura 6.4: Diagrama de base de datos

## 6.5. Tecnologías

### 6.5.1. Nodejs

En esta sección se explicarán las tecnologías utilizadas más relevantes, acompañadas de la justificación de su utilización. Existen varias razones por las cuales se puede elegir Node.js para desarrollar una API para un sistema de registro de usuarios mediante huella dactilar:

Escalabilidad: Node.js es conocido por su capacidad para manejar grandes volúmenes

de tráfico, lo que lo hace ideal para aplicaciones que deben manejar múltiples solicitudes simultáneamente. Esto es especialmente importante para una aplicación de registro de usuarios, que puede tener un alto volumen de solicitudes en momentos de mayor demanda. Actualmente el sistema no requiere grandes prestaciones para funcionar, pero teniendo en cuenta el contexto empresarial en el que se ubica el proyecto creo que es una decisión acertada su utilización.

**Velocidad:** Node.js está construido sobre el motor JavaScript V8 de Google, que está diseñado para ejecutar JavaScript de manera eficiente. Esto lo hace extremadamente rápido en comparación con otros lenguajes de programación, lo que es importante para una aplicación de registro de usuarios, donde la velocidad es esencial para la experiencia del usuario.

**Facilidad de uso:** Node.js es fácil de aprender y utilizar, especialmente si ya estás familiarizado con JavaScript. Además, hay una gran cantidad de módulos y paquetes disponibles en el ecosistema de Node.js que pueden acelerar el proceso de desarrollo.

**Comunidad activa:** Node.js tiene una gran comunidad de desarrolladores que están constantemente creando nuevas herramientas y paquetes para mejorar la plataforma. Esto significa que hay una gran cantidad de recursos disponibles en línea para ayudarte a resolver problemas y desarrollar tu aplicación más rápidamente.

**Integración con tecnologías front-end:** Node.js se integra bien con otras tecnologías de front-end, como Angular y React, lo que significa que puedes construir una aplicación completa desde el back-end hasta el front-end utilizando las mismas herramientas y tecnologías.

En general, en mi opinión Node.js es una excelente opción para desarrollar una API para este sistema debido a su escalabilidad, velocidad, facilidad de uso, comunidad activa y capacidad de integración con tecnologías de front-end.

### 6.5.2. JWT

JWT (JSON Web Token) es una forma segura y eficiente de transmitir información de autenticación y autorización entre el servidor y el cliente. Algunas de las razones por las que considerar JWT en este proyecto son:

**Seguridad:** JWT utiliza una firma digital para garantizar que los datos transmitidos no hayan sido manipulados y que provengan de una fuente confiable. Esto significa que incluso si un tercero intercepta la información de autenticación y autorización, no podrá cambiarla sin ser detectado.

**Escalabilidad:** Como JWT es un formato basado en texto plano, se puede transmitir de manera eficiente a través de cualquier red, lo que lo hace ideal para aplicaciones que manejan grandes volúmenes de tráfico.

**Autenticación y autorización simplificadas:** Con JWT, el servidor puede autenticar al usuario y proporcionar una respuesta que incluya la información de autorización en el token. El cliente puede enviar el token con cada solicitud para acceder a los recursos protegidos sin tener que enviar las credenciales de autenticación cada vez.

**Interoperabilidad:** JWT es un estándar abierto, lo que significa que puede ser utilizado por cualquier lenguaje de programación y cualquier plataforma, lo que lo hace ideal para aplicaciones que están desarrolladas en diferentes tecnologías.

**Menor carga en el servidor:** Al utilizar JWT, el servidor no necesita almacenar ninguna información de autenticación y autorización. Esto significa que hay menos carga en el servidor y se puede escalar de manera más eficiente.

En general, JWT es una buena opción para este proyecto porque proporciona una forma segura y eficiente de transmitir información de autenticación y autorización entre el servidor y el cliente, lo que puede simplificar el proceso de autenticación y autorización y reducir la carga en el servidor.

### 6.5.3. MongoDB

MongoDB es una buena opción para este proyecto por varias razones:

Estructura de datos flexible: MongoDB es una base de datos NoSQL, lo que significa que no utiliza tablas y esquemas fijos. En su lugar, utiliza documentos JSON que se pueden modificar y actualizar de manera flexible. En un proyecto de registro de usuarios mediante huella dactilar, la estructura de los datos puede ser variable y cambiante, por lo que la flexibilidad de MongoDB es una gran ventaja.

Escalabilidad: MongoDB es conocido por su capacidad para escalar horizontalmente, lo que significa que se pueden agregar más servidores para aumentar el rendimiento y la capacidad de la base de datos. Esto es importante en proyectos que pueden experimentar un gran crecimiento y que necesitan una base de datos escalable.

Alto rendimiento: MongoDB está diseñado para trabajar con grandes cantidades de datos y se enfoca en el rendimiento. En proyectos donde se espera un alto tráfico y una gran cantidad de datos, MongoDB puede ofrecer un mejor rendimiento que otras bases de datos.

Integración con Node.js: MongoDB es una base de datos NoSQL popular en la comunidad Node.js. Hay una gran cantidad de módulos y paquetes disponibles para trabajar con MongoDB en Node.js, lo que hace que la integración entre el servidor y la base de datos sea más sencilla.

Funcionalidades avanzadas: MongoDB cuenta con varias funcionalidades avanzadas, como índices geoespaciales y búsqueda de texto completo. En este caso no son utilizadas pero son funcionalidades interesantes para una posible mejora o ampliación del sistema.

En resumen, MongoDB es una buena opción para este proyecto debido a su estructura de datos flexible, escalabilidad, alto rendimiento, integración con Node.js y funcionalidades avanzadas. Todo esto puede contribuir a una base de datos eficiente y fácil de manejar para el proyecto.



# Capítulo 7

## Desarrollo

### 7.1. Descripción de la arquitectura

Para el servidor del sistema desarrollado se ha optado por Una arquitectura de 3 capas. Esta arquitectura es un enfoque común para desarrollar aplicaciones, y puede ser aplicada al desarrollo de servidores en Node.js. En este caso, las tres capas se pueden organizar de la siguiente manera: la capa del controlador, la capa de servicio (o rutas) y la capa de datos (DB).

**Capa del controlador:** La capa del controlador es responsable de recibir las solicitudes del cliente y coordinar las acciones correspondientes. Aquí se manejan las interacciones con el cliente, como la validación de datos de entrada y la autenticación. Los controladores también llaman a los servicios para realizar operaciones más específicas. En Node.js, esto se puede implementar utilizando un enrutador o middleware, como Express.js.

**Capa de servicio (o rutas):** La capa de servicio, también conocida como capa de rutas, se encarga de manejar las operaciones y la lógica de negocio de la aplicación. Aquí se definen las rutas específicas que el servidor puede manejar y se asocian con las funciones correspondientes. Por ejemplo, una ruta GET `/usuarios` estará asociada a una función que recupera todos los usuarios de la base de datos. Estas funciones de servicio pueden interactuar con la capa de datos para realizar operaciones de lectura o escritura en la base

de datos.

**Capa de datos (DB):** La capa de datos es responsable de interactuar con la base de datos o cualquier otro medio de almacenamiento de datos utilizado en la aplicación. Aquí se definen los modelos de datos, se realizan las consultas a la base de datos y se maneja la persistencia de los datos. Es común utilizar un ORM (Object-Relational Mapping) o un ODM (Object-Document Mapping) en Node.js, como Sequelize o Mongoose, respectivamente, para facilitar la interacción con la base de datos.

En resumen, la arquitectura de 3 capas en un servidor Node.js se divide en el controlador, el servicio (o rutas) y la capa de datos. El controlador maneja las interacciones con el cliente, la capa de servicio gestiona las operaciones y la lógica de negocio, y la capa de datos interactúa con la base de datos. Esta estructura ayuda a mantener el código organizado, modular y fácil de mantener.

## 7.2. Configuración del entorno

El entorno es muy importante a la hora de comenzar con el desarrollo de un sistema. Para el sistema de check-in se ha dispuesto un entorno de desarrollo para el servidor y el dispositivo lector (raspberry)

Para comenzar con el desarrollo del servidor es necesario instalar una serie de herramientas que permiten el trabajo con nodejs, entre las que se incluyen los paquetes "npm"(gestor de paquetes), "nvm"(control de versiones de node), "express", "jwt" y "mongoose" entre otros. También se ha utilizado git para el control de versiones, a través del cliente "fork" para mac OS.

En cuanto a la raspberrry, se ha trabajado de la siguiente manera: Este dispositivo ofrece la posibilidad de conectar periféricos como un ordenador normal, pero debido a su rendimiento la navegación por internet (consulta de librerías, stackoverflow, resolución de dudas. . .) se hace lenta y pesada, es por ello que se ha trabajado de forma remota. Para el trabajo en la oficina se ha establecido una dirección IP fija para la Raspberrry, y se ha establecido una conexión ssh para el uso de terminal. Por otro lado se ha utilizado la

herramienta FileZilla para establecer una conexión TCP y solucionar así el problema de la transferencia de archivo.

Es importante destacar el uso vital de la librería "Adafruit fingerprint" que permite la comunicación con el sensor de huellas.

Para todo el desarrollo se ha utilizado el IDE Visual Studio Code, este entorno permite una edición y visualización del código sencillas, gracias a sus múltiples extensiones disponibles para cada lenguaje.

## 7.3. Integración de la base de datos

La base de datos se utiliza para almacenar la información relativa a los empleados de la empresa, así como el respectivo histórico de accesos de cada usuario. En cuanto al desarrollo es importante comentar como se ha integrado su existencia en el proyecto.

La base de datos es modificada solamente por el servidor, a través de la capa de datos mencionada en el apartado 7.1 , esto quiere decir que el servidor, en función de las peticiones recibidas, obtiene, modifica o añade los datos necesarios. La raspberry no puede de forma autónoma modificar esta base de datos, esta decisión es tomada por varios motivos:

- Seguridad: Al evitar que la Raspberry Pi tenga acceso directo a la base de datos, se reduce el riesgo de exposición de información sensible. Si la Raspberry Pi se ve comprometida, un atacante no tendría acceso directo a la base de datos y, por lo tanto, no podría comprometer la integridad de los datos almacenados en ella.
- Escalabilidad y flexibilidad: Al separar la autenticación de la capa de datos, se facilita la escalabilidad del sistema. Si en el futuro se necesita agregar más nodos o servidores para manejar la carga, se puede hacer sin afectar la capa de datos. Además, esto permite cambiar o actualizar la tecnología o la ubicación de la base de datos sin afectar directamente la autenticación.

## 7.4. Desarrollo adicional

Una vez finalizado todo el desarrollo que corresponde al proyecto, y que se estimó en fase de planificación, análisis y diseño, se han llevado a cabo una serie de mejoras entre las que podemos destacar las siguientes:

- Se ha desarrollado una simple interfaz gráfica para el dispositivo, en concreto para las acciones de grabado de huella y de acceso, esto hace que se convierta en un producto más intuitivo y que no sea tan necesaria la presencia del administrador para llevar a cabo estas acciones. Esta interfaz se ha desarrollado mediante la herramienta "tkinter" de python.
- Se ha desplegado el proyecto en un servidor en la nube, lo que incrementa la versatilidad del dispositivo, ya que puede conectarse desde cualquier red sin configuración adicional. Esto se ha realizado mediante la herramienta docker por la parte del desarrollo. La infraestructura y el mantenimiento del servidor son suministrados por Brooktec S.L.
- Se ha diseñado una pieza 3d mediante el programa "catia" que sirve como pieza de unión entre la raspberry y el sensor de huellas, así como del led informativo que se ha incorporado también para mejorar la experiencia de usuario.

# Capítulo 8

## Pruebas y validación

En el apartado de pruebas y validación, se lleva a cabo un conjunto de actividades destinadas a verificar y validar el funcionamiento correcto y la calidad del sistema desarrollado. Estas pruebas y validaciones son esenciales para garantizar que el software cumpla con los requisitos establecidos y funcione de acuerdo con las expectativas del cliente.

La fase de pruebas y validación implica la ejecución de diferentes tipos de pruebas para evaluar aspectos específicos del sistema. Estas pruebas permiten identificar posibles errores o deficiencias en el software, brindando la oportunidad de corregirlos antes de que el sistema se implemente en un entorno de producción.

Las pruebas pueden incluir desde pruebas unitarias, donde se evalúa el correcto funcionamiento de unidades individuales de código, hasta pruebas de integración, donde se verifica la correcta interacción entre los diferentes componentes del sistema.

Además de las pruebas técnicas, la validación implica la participación del cliente o usuario final para evaluar el sistema desde su perspectiva y verificar si cumple con sus requisitos y necesidades. Estas validaciones pueden incluir revisiones y pruebas de aceptación por parte del cliente, donde se verifica que el sistema se ajuste a los criterios de aceptación previamente establecidos.

En resumen, la fase de pruebas y validación es fundamental para asegurar la calidad del software desarrollado. A través de diversas pruebas técnicas y validaciones por parte del cliente, se verifica el cumplimiento de los requisitos y se identifican y corrigen posibles

problemas antes de la implementación en producción. Esto contribuye a ofrecer un sistema confiable, funcional y satisfactorio para los usuarios finales.

Se diferencian dos apartados, pruebas automáticas, y pruebas con usuarios finales:

## 8.1. Pruebas automáticas

Junto al desarrollo propiamente dicho del servidor, se ha implementado un sistema de pruebas automáticas sobre el código del mismo gracias a la tecnología "jest". Este sistema consta de pruebas unitarias así como de pruebas de integración. Uno de los objetivos indicado en el requisito no funcional RNFA08 es el lograr una cobertura total del código elaborado, este requisito se satisface mediante estas pruebas.

## 8.2. Pruebas con usuarios finales

Una vez desarrollada una primera iteración del producto final, se ha llevado a cabo una prueba con usuarios finales. La prueba en cuestión consistió en el registro de la información de los empleados de la oficina que se encontraban en ella el día que se realizó la mencionada prueba. La descripción y los resultados de esta se detallan a continuación:

### Introducción

El presente informe documenta una prueba realizada para registrar las huellas dactilares de los de la oficina de Valladolid de Brooktec. Se describe la metodología, se presentan los resultados y se extraen conclusiones relevantes para el proyecto. El registro preciso y seguro de las huellas dactilares es fundamental para mejorar el control de acceso y garantizar la seguridad en las instalaciones de la empresa.

## Metodología

Para llevar a cabo esta prueba se ha utilizado el sistema desarrollado, compuesto por la Raspberry Pi y el sensor biométrico por un lado, y el servidor con la base de datos por el otro. El proceso seguido ha sido el siguiente, repetido por cada empleado seleccionado:

- Arrancar el servidor y el programa de gestión de huellas en la raspberry.
- Solicitar la opción de registrar huella
- Introducir manualmente el nombre de usuario y el deviceId
- Solicitar al empleado que introduzca la huella dos veces
- Comprobar que el proceso ha sido satisfactorio y el usuario aparece registrado en la base de datos

## Resultados

En la prueba se llevo a cabo el proceso con 10 empleados, incluido el alumno. De estos 10 alumnos la prueba tuvo un resultado completamente satisfactorio para 8 de ellos, presentando un problema en dos de los casos, es decir, un 80 por ciento de acierto.

El problema que tuvo lugar es un problema de uso normal, relativo a la colocación del dedo en el sensor, ya que si no se coloca de una manera correcta en la primera toma, al realizar la segunda el sistema comunicará que las tomas no coinciden.

Es necesario mencionar que posteriormente al registro, se probó a lanzar el programa de reconocimiento de huellas en la raspberry, obteniendo un resultado negativo, ya que aparentemente no se registró ninguna entrada en el histórico de accesos. Este problema ha sido registrado y se solucionará para la siguiente prueba.

También se ha observado que actualmente es necesario introducir manualmente el deviceId, y si se introduce uno que ya existe se sobrescribirá el usuario con ese identificador. Se ha registrado el problema y se solucionará para la siguiente prueba.

### 8.2.1. Segunda iteración

Tras analizar los resultados de la prueba realizada, se han solucionado los fallos detectados y realizada una segunda prueba, de la que no se adjunta informe, pero en la que se pudo comprobar como todos los inconvenientes de la primera prueba no sucedieron de la misma manera en la segunda, obteniendo así el resultado deseado en un principio.

# Capítulo 9

## Resultados

### 9.1. Generales

En primer lugar procede comentar el correcto desarrollo del proyecto. El producto planificado se ha llevado a cabo en su totalidad consiguiendo los objetivos propuestos. Esto como no puede ser de otra manera ha generado ciertas dificultades, como pueden ser:

- La correcta integración de la librería necesaria para utilizar el sensor biométrico con la raspberry a la hora de conseguir la funcionalidad deseada.
- El desarrollo de una API desde cero, siendo la primera vez que se efectuaba este proceso.
- Lograr el completo éxito de los tests (pruebas automáticas)
- Establecer un baremo/límite para incorporar mejoras.

En resumen, se consigue como resultado un producto funcional y que cumple con lo que se planteó en el inicio.

## 9.2. Requisitos

En segundo lugar, es necesario mencionar como el producto desarrollado realiza las acciones deseadas/planificadas y cumple con los requisitos establecidos. Podemos destacar lo siguiente:

- Se ha utilizado la raspberry correctamente para la parte de identificación biométrica del usuario.
- Se han utilizado las tecnologías preestablecidas y cumplido con los estándares de calidad de la empresa.
- El código del servidor pasa las pruebas tanto de corrección como los tests elaborados sobre su funcionamiento

En resumen, el proyecto cumple con todos y cada uno de los requisitos establecidos en la fase de planificación, ya sean funcionales o no funcionales

## 9.3. Riesgos

En cuanto a los riesgos se puede destacar la materialización de ellos, el RSG03 y el RSG04:

- La estimación para algunas tareas ha resultado ser algo sesgada y el proyecto ha sido terminado más tarde de lo previsto, pero gracias a las acciones de mitigación, en concreto los sprints de refuerzo, se ha podido terminar según la planificación.
- El tutor ha mostrado una gran disponibilidad, pero es cierto que sobre todo al final, ha habido algún problema a la hora de concretar alguna reunión que ha ralentizado levemente el correcto desarrollo del problema, de nuevo gracias a seguir las pautas de las acciones de mitigación y corrección ha sido posible llevar el proyecto a término.

Una vez mitigados/corregidos estos riesgos, llegando al final del proyecto los riesgos se pueden considerar cerrados.

## 9.4. Calendario

Con respecto al calendario, como se ha mencionado antes se ha trabajado con un margen amplio para evitar problemas de falta de tiempo y se ha conseguido terminar el proyecto junto con el documento de memoria antes de la fecha límite establecida por la universidad.

## 9.5. Presupuesto

No ha habido ningún problema con esta sección a lo largo del proyecto, se ha cumplido el presupuesto planificado sin mayor problema.



# Capítulo 10

## Conclusiones

Las conclusiones del proyecto son las siguientes:

- Se ha logrado mejorar la eficiencia y la precisión del proceso de control de acceso. Los resultados de las pruebas han demostrado una verificación de identidad más rápida, precisa y segura en comparación con los métodos tradicionales. Esto ha permitido agilizar el acceso de los empleados a las instalaciones de la empresa, ahorrando tiempo y mejorando la experiencia del usuario.
- El objetivo de simplificar el proceso de control de acceso se ha alcanzado con éxito. La eliminación del control manual ha reducido la posibilidad de errores humanos y ha optimizado los recursos de la empresa. Los empleados pueden acceder de manera más rápida y conveniente al utilizar su huella dactilar como método de autenticación.
- La solución ha proporcionado una administración centralizada de la información biométrica de los empleados y la gestión del tiempo. La integración del sistema de check-in con el sistema de gestión de recursos humanos facilitará la administración de los datos de los empleados y el seguimiento del tiempo trabajado. Esto permitirá una mejor organización y un mayor control de la gestión del personal.
- La implementación del sistema de check-in biométrico ha mejorado significativamente la seguridad de las instalaciones de la empresa. La tecnología biométrica de

huellas dactilares ofrece un nivel de seguridad mucho más alto en comparación con otros métodos de autenticación.

En resumen, este proyecto ha logrado cumplir satisfactoriamente los objetivos establecidos . Los resultados obtenidos demuestran mejoras en la eficiencia, la seguridad y la gestión del tiempo en la empresa, respaldando la utilidad y efectividad de la solución propuesta.

### **10.1. Siguietes pasos:**

Una vez desarrollado este producto, las aplicaciones son diversas, pero puede enfocarse hacia la correcta conexión del servidor creado con la plataforma de la empresa, a fin de automatizar todavía más el proceso en el que se ha trabajado durante este proyecto. También cabe destacar que la solución es completa y versátil, por lo que podría utilizarse para otras empresas, obteniendo un mayor beneficio de ella.

Parte II

Apéndices



# Apéndice A

## Anexos

### A.1. Información complementaria

#### A.1.1. Especificaciones sensor biométrico

##### Características

- Voltaje de alimentación: DC 3.3V
- Corriente de alimentación: Corriente de funcionamiento: <60mA
- Corriente máxima: <60mA
- Tiempo de entrada de imagen de huella digital: <1.0 segundos
- Área de la ventana: 15.3 18.2 mm
- Resolución: 500dpi

##### Notas

La característica extraída por el algoritmo representa la información de la huella digital. El almacenamiento de huellas digitales, la comparación y la búsqueda se realizan mediante la manipulación de las características de estas.

El procesamiento de huellas dactilares consta de dos procesos: proceso de inicio de sesión y proceso de comparación [donde la coincidencia se divide en comparación de huellas (1: 1) y búsqueda de huellas (1: N)].

Cuando se registra la huella digital, cada huella digital se ingresa dos veces, y las imágenes grabadas se procesan dos veces, y la plantilla compuesta se almacena en el módulo.

Cuando la huella digital coincide, la imagen de la huella digital se ingresa, se verifica y procesa, y luego se combina con la plantilla en el módulo. Si la plantilla coincide con una plantilla especificada en el módulo, se denomina método de comparación, es decir, 1: 1. Si coincide con varias plantillas, es llamado modo de búsqueda (es decir, modo 1: N), el módulo da el resultado correspondiente (pase o falla).

### A.1.2. Especificaciones técnicas raspberry

Característica	Especificación
Procesador	Broadcom BCM2711, Quad-Core Cortex-A72
Velocidad del procesador	1.5 GHz
Memoria RAM	2 GB LPDDR4 SDRAM
GPU	VideoCore VI
Almacenamiento	Ranura para tarjeta microSD
Conectividad	2 puertos USB 3.0, 2 puertos USB 2.0
	2 puertos micro-HDMI (hasta 4Kp60)
	Puerto Ethernet Gigabit
	Wi-Fi 802.11ac de doble banda
	Bluetooth 5.0
Puertos	2 puertos USB 3.0, 2 puertos USB 2.0
	Puerto Ethernet
	Puerto de audio/video de 3.5 mm
	Conector GPIO de 40 pines
	Conector CSI para cámara
	Conector DSI para pantalla
Dimensiones	88 mm x 58 mm x 19.5 mm
Peso	46 gramos

### A.1.3. Interfaz gráfica

A continuación se adjuntan algunas capturas de la interfaz gráfica elaborada para la raspberry.



Figura A.1: Pantalla principal

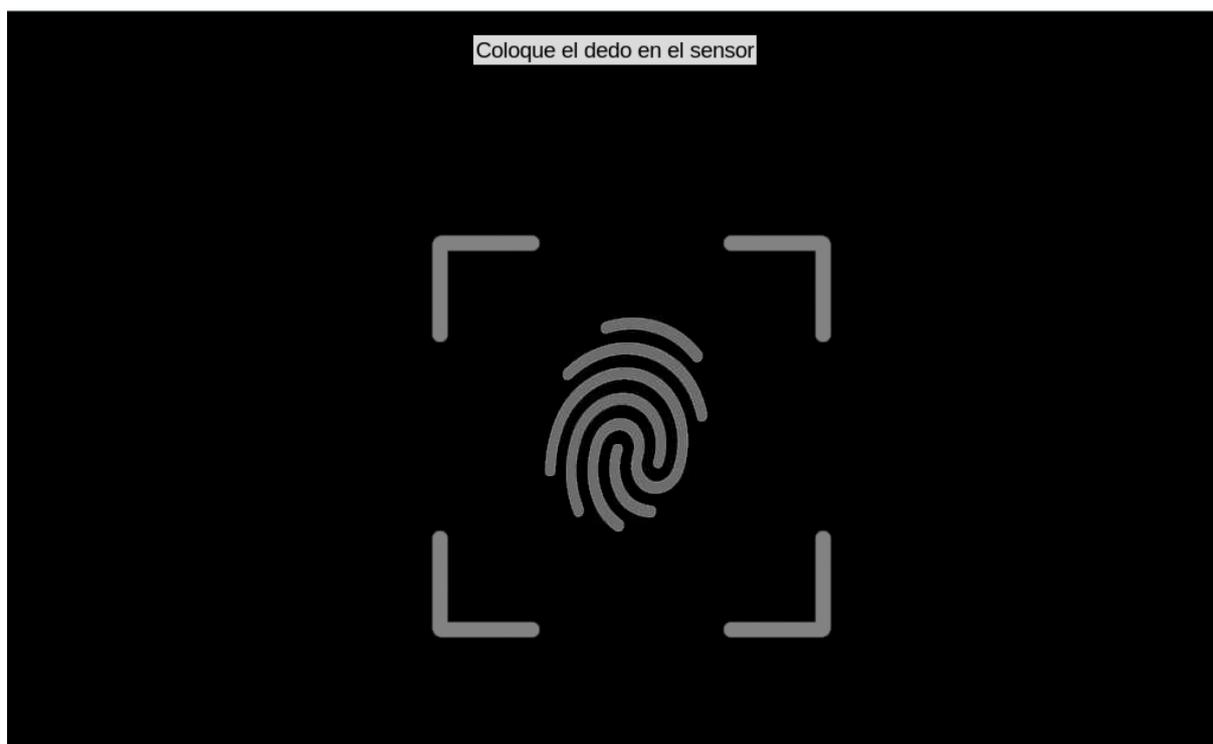


Figura A.2: Pantalla de detección de huella

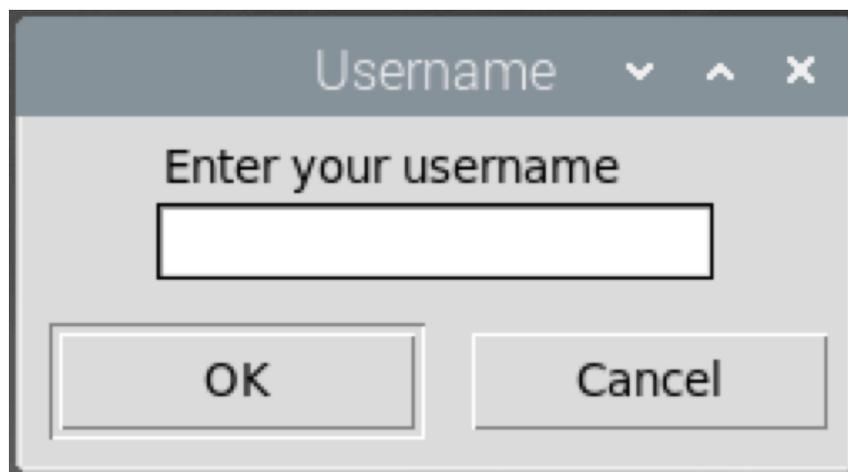


Figura A.3: Pantalla de introducción de nombre de usuario

### A.1.4. Diseño de la pieza 3D

A continuación se pueden ver los planos que se han elaborado para construir la pieza 3d utilizada en el proyecto:

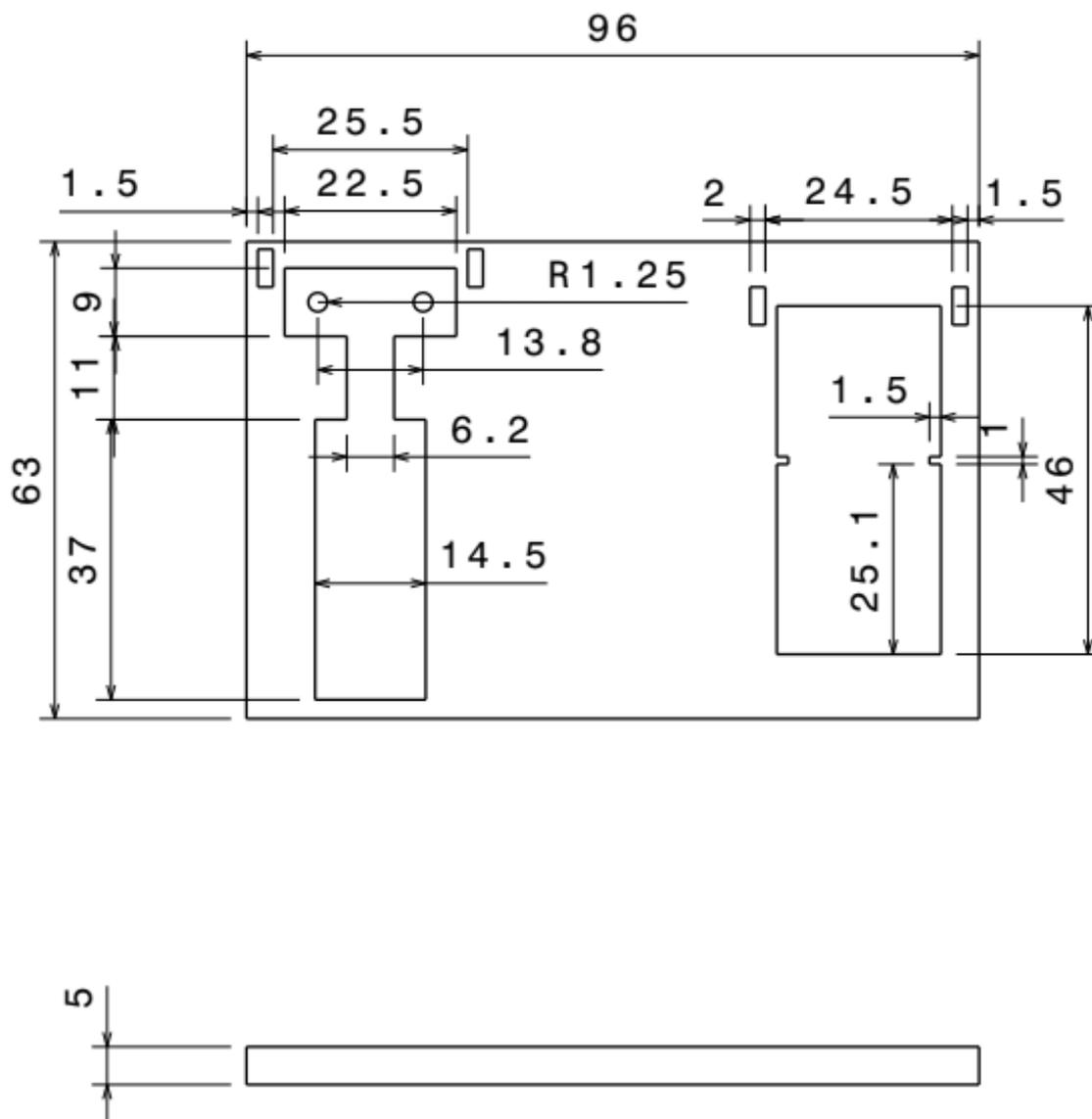


Figura A.4: Vista superior del conjunto tapa-sensor-led

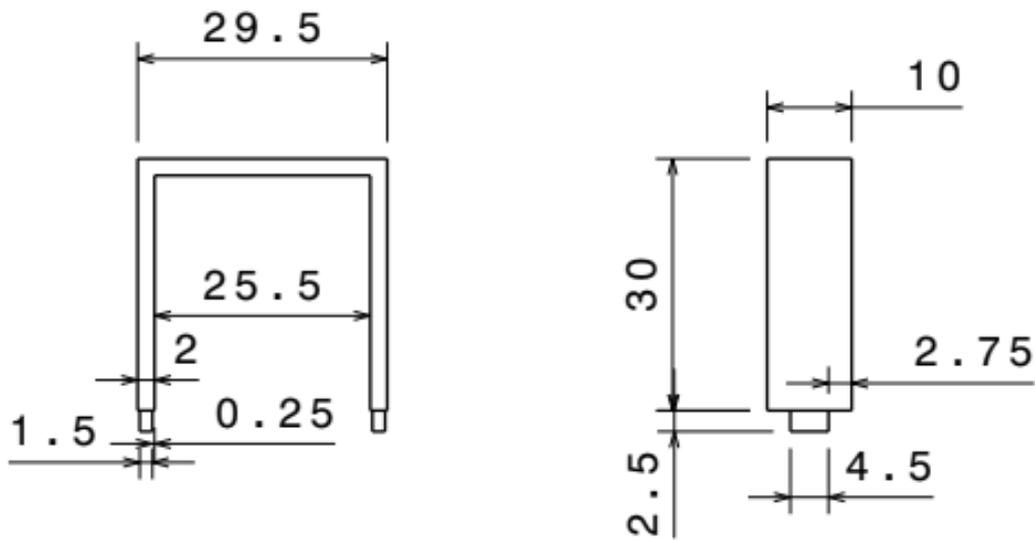


Figura A.5: Vista de las protecciones

### A.1.5. Links Adicionales

A continuación se encuentra el link al repositorio de bitbucket donde se ubica el código fuente de la solución:

<https://bitbucket.org/jorgeandresbrooktec/brooktec-touchpass/src>



# Bibliografía

- [1] *adafruitFingerprint 2014; Adafruit Fingerprint Library 1.0 documentation*. <https://docs.circuitpython.org/projects/fingerprint/en/latest/api.html>.
- [2] Karla Cevallos. *25. UML: Diagrama de Secuencia – INGENIERÍA DEL SOFTWARE*. 2015.
- [3] *Expect · Jest — jestjs.io*. <https://jestjs.io/es-ES/docs/expect#tomatchregexpp-string>.
- [4] *Express application generator — expressjs.com*. <https://expressjs.com/en/starter/generator.html>.
- [5] *How to use Rest-APIs with Raspberry Pi , ESP32 Web Applications for IOT Projects — youtube.com*. <https://www.youtube.com/watch?v=K60ACu4j2Kw>.
- [6] Ionos. *Diagrama de estados*. Jul. de 2020. URL: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-estado-uml/>.
- [7] Esteban Knöbl. *Objetivos SMART: qué son y cómo utilizarlos*. 2018.
- [8] *Lectores de Huellas Dactilares JM-101 - sensor óptico — solectroshop.com*. <https://solectroshop.com/es/lectores-de-huellas-dactilares/5240-lectores-de-huellas-dactilares-jm-101-sensor-optico.html>.
- [9] *Login + JWT | Node.js — bluweb.github.io*. <https://bluweb.github.io/node/07-jwt/#ruta-protegida>.
- [10] Corporation Microsoft. “Diagramas de Casos de Uso”. En: *Microsoft, Corporation* (2015).

- [11] Francisco Coll Morales. *Recursos humanos*. Mar. de 2020. URL: <https://economipedia.com/definiciones/recursos-humanos-rrhh.html>.
- [12] *Node.js MongoDB Create Database* — *w3schools.com*. [https://www.w3schools.com/nodejs/nodejs\\_mongodb\\_create\\_db.asp](https://www.w3schools.com/nodejs/nodejs_mongodb_create_db.asp).
- [13] Armando Díaz Olivera e Idalberto Matamoros Hernández. “El análisis DAFO y los objetivos estratégicos.” En: *Contribuciones a la Economía* (2011).
- [14] Myriam Quiroa. *Recursos Materiales*. Abr. de 2020. URL: <https://economipedia.com/definiciones/recursos-materiales.html>.
- [15] *supertest* — *npmjs.com*. <https://www.npmjs.com/package/supertest>.
- [16] *Usando un lector de huellas dactilares en Raspberry Pi* | *Blog Bujarra.com* — *bujarra.com*. <https://www.bujarra.com/usando-un-lector-de-huellas-dactilares-en-raspberry-pi/>.