



UNIVERSIDADE
LUSÓFONA

OP - Opening Doors

Trabalho Final de Curso

Relatório Intercalar 1º Semestre

Guilherme Ribeiro
João Pedro Carvalho
Daniel Fernandes

Trabalho Final de Curso | LEI | 03 / 12 / 2024

www.ulusofona.pt

Direitos de cópia

OP - Opening Doors, Copyright de Guilherme Ribeiro, Universidade Lusófona.
A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona (UL) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Este documento foi gerado com o processador (pdf/Xe/Lua) \LaTeX e o modelo ULThesis (v1.0.0) [1].

Capítulo 1

Introdução

1.1 Resumo

Este projeto centra-se no desenvolvimento de um sistema web de controlo remoto de portas, utilizando tecnologias modernas como Flask [2] para o backend e SQLite [3] para a gestão de dados. A principal motivação deste trabalho foi atender à crescente demanda por soluções práticas e seguras para a automatização de acessos em ambientes residenciais e corporativos. O sistema implementado permite registar utilizadores, monitorizar estados das portas e gerir as suas ações de forma dinâmica, alterando entre os estados "aberta" e "fechada".

A aplicação foi desenvolvida com uma arquitetura bem definida, incluindo endpoints RESTful para comunicação, validação de dados e segurança no acesso às operações. Durante o processo, foram realizadas implementações iterativas, testes contínuos no Postman, e integração de uma base de dados eficiente que armazena informações das portas e dos utilizadores.

Este trabalho contribui com uma solução tecnológica para problemas práticos do quotidiano, destacando-se pela flexibilidade, fiabilidade e potencial de integração com sistemas IoT futuros. Além disso, demonstra como tecnologias acessíveis podem ser utilizadas para resolver desafios reais, promovendo inovação e segurança.

1.2 Motivação

Com o aumento da digitalização em sistemas de acesso, a necessidade de soluções seguras e eficientes para o controlo remoto de dispositivos, como portas, tornou-se evidente. A motivação para este projeto surge da possibilidade de criar uma aplicação prática que integre tecnologia moderna, contribuindo para a automatização de tarefas quotidianas e para a segurança em contextos residenciais ou empresariais.

1.3 Identificação do Problema

A maioria das smart locks disponíveis no mercado foi desenvolvida para uso pessoal, focando-se em residências individuais. Estas trancas são ótimas para um único utilizador ou pequenas famílias, mas tornam-se limitadas e ineficazes em condomínios com muitos moradores. Não conseguem gerir múltiplos utilizadores, controlar o acesso a diferentes portas ou registar eventos como quem entrou e a que horas. A falta de uma solução centralizada impede que administradores, como seguranças, possam gerir facilmente as permissões de acesso em tempo real ou garantir a segurança de todos os residentes. A nossa smart lock para condomínios resolve esse problema, ao ser projetada para gerir o acesso de muitos utilizadores de forma eficiente e segura. Permite que administradores adicionem ou removam moradores, atribuam portas específicas, monitorizem acessos e enviem notificações automáticas. Com um sistema centralizado e escalável, garante que cada morador tem acesso apenas à sua residência, enquanto os administradores mantêm o controlo completo da segurança do condomínio.

1.3.1 Introdução

- Apresenta o contexto e os objetivos do projeto, explicando a relevância do sistema proposto.
- Detalha a motivação e identifica os problemas que o projeto se propõe a resolver, como a necessidade de soluções de controlo de acesso mais seguras e escaláveis em comunidades residenciais.
- Explica a estrutura do relatório e o que será abordado em cada secção, oferecendo uma visão geral dos componentes do sistema e do processo de desenvolvimento.

1.3.2 Benchmarking

- Analisa soluções semelhantes disponíveis no mercado, avaliando as suas funcionalidades, vantagens e limitações.
- Destaca os aspetos técnicos e de design que podem ser incorporados ou melhorados no sistema *OP - Opening Doors*.
- Estabelece critérios de comparação e justifica as escolhas feitas para o desenvolvimento do projeto.

1.3.3 Viabilidade e Pertinência

- Avalia a viabilidade técnica do projeto, considerando a implementação de componentes como o servidor Flask, integração com smart locks e uso de bases de dados SQLite.
- Justifica a pertinência do projeto no contexto de comunidades residenciais, abordando questões como segurança, eficiência e facilidade de uso.
- Apresenta os recursos necessários, incluindo hardware, software e tempo, para garantir a execução bem-sucedida do projeto.

1.3.4 Solução Proposta

- Detalha a arquitetura do sistema, destacando a separação entre o frontend e o backend, e a integração com dispositivos físicos como smart locks.
- Apresenta os principais requisitos funcionais, organizados por prioridades (*Must*, *Should* e *Could*), e explica o papel de cada funcionalidade no sistema.
- Explica o fluxo de dados entre os utilizadores, o sistema e os dispositivos físicos, conforme representado no fluxograma de interação.

1.3.5 Calendário

- Apresenta o cronograma Gantt que detalha as diferentes etapas do projeto, desde a investigação inicial até à implementação final.
- Explica as atividades principais, como a montagem do Arduino, desenvolvimento do servidor Flask e criação do protótipo, destacando a sequência lógica entre elas.
- Discute a importância do planeamento para garantir a execução eficiente e dentro dos prazos estabelecidos.

Capítulo 2

Abstract

2.1 Summary

This project focuses on the development of a web-based remote door control system using modern technologies such as Flask [2] for the backend and SQLite [3] for data management. The primary motivation for this work was to meet the growing demand for practical and secure solutions for automating access in residential and corporate environments. The implemented system allows for user registration, door status monitoring, and dynamic management of their actions, toggling between "open" and "closed" states.

The application was developed with a well-defined architecture, including RESTful endpoints for communication, data validation, and secure access to operations. During the process, iterative implementations were carried out, with continuous testing using Postman, and the integration of an efficient database to store door and user information.

This work contributes a technological solution to practical everyday problems, standing out for its flexibility, reliability, and potential for integration with future IoT systems. Furthermore, it demonstrates how accessible technologies can be used to address real-world challenges, promoting innovation and security.

2.2 Motivation

With the rise of digitalization in access control systems, the need for secure and efficient solutions for remotely controlling devices such as doors has become evident. The motivation for this project arises from the opportunity to create a practical application that integrates modern technology, contributing to the automation of daily tasks and enhancing security in residential or business contexts.

2.3 Problem Identification

Most smart locks available in the market are designed for personal use, focusing on individual residences. These locks are excellent for single users or small families but become limited and ineffective in condominiums with many residents. They fail to manage multiple users, control access to different doors, or log events such as who entered and when. The lack of a centralized solution prevents administrators, such as security personnel, from easily managing access permissions in real time or ensuring the safety of all residents.

Our smart lock for condominiums addresses this issue by being designed to efficiently and securely manage access for many users. It allows administrators to add or remove residents, assign specific doors, monitor access, and send automatic notifications. With a centralized and scalable system, it ensures that each resident has access only to their unit while administrators maintain complete control over the condominium's security.

2.3.1 Introduction

- Presents the context and objectives of the project, explaining the relevance of the proposed system.

- Details the motivation and identifies the problems the project aims to solve, such as the need for safer and more scalable access control solutions in residential communities.
- Explains the structure of the report and what will be addressed in each section, providing an overview of the system components and the development process.

2.3.2 Benchmarking

- Analyzes similar solutions available in the market, evaluating their functionalities, advantages, and limitations.
- Highlights technical and design aspects that can be incorporated or improved in the *OP - Opening Doors* system.
- Establishes comparison criteria and justifies the choices made for the project development.

2.3.3 Feasibility and Relevance

- Evaluates the technical feasibility of the project, considering the implementation of components such as the Flask server, smart lock integration, and the use of SQLite databases.
- Justifies the project's relevance in the context of residential communities, addressing issues such as security, efficiency, and ease of use.
- Presents the necessary resources, including hardware, software, and time, to ensure the successful execution of the project.

2.3.4 Proposed Solution

- Details the system architecture, emphasizing the separation between frontend and backend and the integration with physical devices like smart locks.
- Presents the main functional requirements, organized by priorities (*Must*, *Should*, and *Could*), and explains the role of each functionality in the system.
- Explains the data flow between users, the system, and physical devices, as represented in the interaction flowchart.

2.3.5 Schedule

- Presents the Gantt chart that details the different stages of the project, from initial research to final implementation.
- Explains the main activities, such as Arduino assembly, Flask server development, and prototype creation, highlighting the logical sequence between them.
- Discusses the importance of planning to ensure efficient execution within the established deadlines.

Conteúdo

1 Introdução	2
Resumo	2
1.1 Resumo	2
1.2 Motivação	2
1.3 Identificação do Problema	2
1.3.1 Introdução	3
1.3.2 Benchmarking	3
1.3.3 Viabilidade e Pertinência	3
1.3.4 Solução Proposta	3
1.3.5 Calendário	3
2 Abstract	4
Abstract	4
2.1 Summary	4
2.2 Motivation	4
2.3 Problem Identification	4
2.3.1 Introduction	4
2.3.2 Benchmarking	5
2.3.3 Feasibility and Relevance	5
2.3.4 Proposed Solution	5
2.3.5 Schedule	5
Conteúdo	6
Lista de Figuras	8
Lista de Tabelas	9
3 Benchmarking	10
3.1 Identificação de Soluções no Mercado	10
3.2 Relação Front-end to Back-end	11
4 Viabilidade e Pertinência	13
4.1 Viabilidade	13
4.2 Pertinência	13
5 Solução Proposta	15
5.1 Tecnologias Utilizadas	15
5.2 Aplicação de Disciplinas do Curso	15
5.3 Requisitos Funcionais	16
5.3.1 Contexto adicional	16
5.3.2 Tabela de Requisitos Funcionais	16
6 Calendário	18
6.1 Calendário	18

7 Conclusão	20
Bibliografia	21

Lista de Figuras

3.1	Fluxograma Front-end to Back-end	12
6.1	Diagrama de Gantt	19

Lista de Tabelas

3.1	Tabela de Comparação (Parte 1)	10
3.2	Tabela de Comparação (Parte 2)	10
3.3	Tabela de Comparação (Parte 3)	11
5.1	Tabela de Requisitos Funcionais do OP - Opening Doors	17

Capítulo 3

Benchmarking

Ao desenvolvermos a solução proposta para o sistema de smart lock para condomínios, identificámos várias soluções semelhantes já disponíveis no mercado. Como parte essencial deste processo, realizámos uma análise comparativa com alternativas como a August Wi-Fi Smart Lock [4], Nuki Smart Lock [5], e outras. Esta comparação permite-nos avaliar o nível de inovação da nossa solução, bem como compreender as melhores práticas na gestão de acessos, segurança e administração de múltiplos utilizadores em ambientes residenciais complexos, como condomínios.

3.1 Identificação de Soluções no Mercado

Abaixo estão as tabelas(3.1 , 3.2 , 3.3) que exemplificam as diferenças em funcionalidades entre várias soluções de smart locks disponíveis no mercado, incluindo o OP - Opening Doors, desenvolvido neste projeto. Estas tabelas permitem uma análise comparativa clara, destacando as funcionalidades oferecidas por cada dispositivo, como registo de acesso, notificações, gestão de utilizadores e sistemas de travamento automático. Por exemplo, observa-se que o OP - Opening Doors fornece funcionalidades essenciais como registo de acesso e notificações, posicionando-se como uma solução flexível e eficiente. A análise destas tabelas reforça o alinhamento do projeto com as necessidades modernas de automatização de acessos

Funcionalidades	August Wi-Fi Smart Lock	Nuki Smart Lock
Registo de Acesso	X	X
Notificações de Quando a Porta Foi Aberta	X	X
Abrir Porta Através de Aplicativo	X	X
Gerir Usuários (Adicionar Contas)	X	X
Auto-lock (Trava Automática)	X	X
Temporizador para Auto-lock	—	X
Gestor de multiplas Locks	—	—
Alarme de porta forçada	—	—

Tabela 3.1: Tabela de Comparação (Parte 1)

Funcionalidades	Yale Fechadura Inteligente	Schlage Encode
Registo de Acesso	X	X
Notificações de Quando a Porta Foi Aberta	X	X
Abrir Porta Através de Aplicativo	X	X
Gerir Usuários (Adicionar Contas)	X	X
Auto-lock (Trava Automática)	X	X
Temporizador para Auto-lock	X	X
Gestor de multiplas Locks	—	—
Alarme de porta forçada	—	—

Tabela 3.2: Tabela de Comparação (Parte 2)

Funcionalidades	Lockly Secure Pro	OP - Opening Doors
Registro de Acesso	X	X
Notificações de Quando a Porta Foi Aberta	X	X
Abrir Porta Através de Aplicativo	X	X
Gerir Usuários (Adicionar Contas)	X	X
Auto-lock (Trava Automática)	X	X
Temporizador para Auto-lock	X	X
Gestor de multiplas Locks	—	X
Alarme de porta forçada	—	X

Tabela 3.3: Tabela de Comparação (Parte 3)

3.2 Relação Front-end to Back-end

Nos sistemas modernos de controlo de acesso, a integração entre frontend e backend é essencial para garantir a comunicação eficiente e segura entre os utilizadores e os dispositivos físicos, como smart locks. Este fluxo de dados e comandos é mediado por diferentes camadas tecnológicas, que colaboram para oferecer uma experiência intuitiva, confiável e escalável.

O fluxograma abaixo ilustra a relação entre as diferentes partes do sistema no projeto OP - Opening Doors, destacando o papel de cada componente:

- O usuário é o ponto de partida, interagindo com o sistema através de uma interface gráfica acessível e responsiva, desenvolvida no frontend.
- O frontend recebe os pedidos do utilizador, como abrir ou fechar uma porta, e traduz esses comandos para o backend, utilizando chamadas API no formato REST.
- O backend, implementado com ferramentas como Flask, processa a lógica do sistema e comunica com a base de dados para consultar ou atualizar informações, como permissões de acesso ou registos de utilizadores.
- A base de dados, representada por um sistema SQLite, armazena de forma segura todos os dados necessários, incluindo registos de acessos, informações sobre utilizadores e configurações de dispositivos.
- Por fim, o backend retorna os resultados ao frontend, que apresenta as respostas ou ações ao utilizador de forma clara e informativa.

Este modelo garante uma separação de responsabilidades, facilitando a manutenção, escalabilidade e segurança do sistema. Além disso, a arquitetura modular permite futuras integrações com novos dispositivos ou serviços externos, promovendo a evolução contínua da solução.

O fluxograma fornece uma visão clara desse fluxo de interação, permitindo compreender como as diferentes partes se interligam para formar um sistema coeso e eficiente.

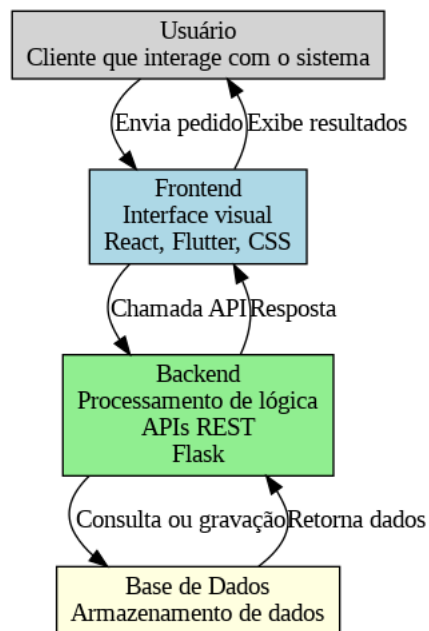


Figura 3.1: Fluxograma Front-end to Back-end

Capítulo 4

Viabilidade e Pertinência

4.1 Viabilidade

O OP - Opening Doors apresenta-se como uma solução tecnologicamente viável, dado o seu foco em atender às necessidades específicas de grandes comunidades residenciais e condomínios. A sua conceção foi idealizada para operar numa escala superior às "smart locks" convencionais, como a August Wi-Fi Smart Lock [4] e a Nuki Smart Lock [5]. Estas soluções existentes, embora eficazes para residências individuais, enfrentam limitações significativas quando aplicadas em contextos de larga escala.

Diferentemente dessas alternativas, o OP - Opening Doors destaca-se por:

- **Gestão de utilizadores em tempo real:** Permite a criação, edição e remoção de acessos de forma centralizada, adaptando-se a diferentes perfis e necessidades dentro da comunidade.
- **Atribuição de permissões específicas:** Oferece um sistema granular de permissões que possibilita controlar quem pode aceder a áreas específicas e em quais horários.
- **Monitorização de acessos:** Proporciona relatórios detalhados e históricos de acessos, garantindo maior segurança e rastreabilidade.
- **Escalabilidade:** Projetado para suportar centenas de utilizadores simultaneamente, sem comprometer o desempenho.

Este nível de inovação demonstra a capacidade do projeto para superar as limitações das tecnologias existentes, tornando-se um produto comercialmente viável e competitivo. Adicionalmente, o projeto tem potencial de expansão para mercados corporativos e residenciais de grande escala, assegurando continuidade e relevância mesmo após a conclusão do Trabalho Final de Curso (TFC). A implementação de tecnologias modernas, como Flask [2] e serviços de Cloud Computing, reforça a robustez e confiabilidade da solução.

4.2 Pertinência

A pertinência do OP - Opening Doors é evidente na sua capacidade de abordar um problema real e crescente no mercado: a necessidade de gestão eficiente de acessos em ambientes coletivos. Enquanto as "smart locks" convencionais, como a August Wi-Fi Smart Lock [4] e a Nuki Smart Lock [5], são desenhadas para uso individual ou em pequena escala, esta solução preenche uma lacuna significativa ao oferecer uma abordagem personalizada e centralizada, essencial para grandes comunidades residenciais.

Os principais diferenciais do projeto incluem:

- **Eficiência:** Simplifica a gestão de acessos, reduzindo o tempo e os recursos necessários para gerir entradas em comunidades de grande porte.

- **Segurança:** Com monitorização em tempo real e permissões específicas, garante maior controlo e proteção contra acessos não autorizados.
- **Flexibilidade:** Adapta-se a diferentes necessidades e configurações, oferecendo personalização para ambientes residenciais ou corporativos.

Ao focar-se na eficiência, segurança e flexibilidade de utilização, o projeto apresenta um impacto positivo e direto, contribuindo para melhorar a qualidade de vida e gestão em condomínios e outros ambientes coletivos. Com base no estado atual da tecnologia e nas tendências de mercado, o OP - Opening Doors demonstra o seu potencial para estabelecer novos padrões no setor, fortalecendo a sua relevância e capacidade de inovação.

Capítulo 5

Solução Proposta

O projeto OP - Opening Doors visa criar uma solução de smart lock inovadora, especialmente desenhada para ser utilizada em condomínios e outros ambientes com múltiplos utilizadores. A proposta é não só oferecer um sistema seguro e eficiente para o controlo de acessos, mas também desenvolver uma plataforma escalável que permita a integração de diferentes dispositivos e a gestão de múltiplos utilizadores de forma simples. Abaixo, são descritas as principais tecnologias e fundamentos para a construção desta solução.

5.1 Tecnologias Utilizadas

- **Internet das Coisas (IoT) e Smart Locks:** A utilização de dispositivos IoT [6] é fundamental para garantir a interatividade entre as smart locks e a plataforma. Os dispositivos IoT permitem que os utilizadores possam controlar as fechaduras remotamente, proporcionando uma experiência segura e prática.
- **Aplicações Web e Móvel:** A plataforma será acessível através de uma aplicação web e móvel, desenvolvidas com as tecnologias React [7] (para o front-end) e Node.js [8] (para o back-end). Isso permitirá uma interface de utilizador fluida e de fácil utilização, tanto para gestores de condomínios quanto para os residentes.
- **Arquitetura de Microserviços:** A escolha da arquitetura de microserviços permite que a aplicação seja modular e escalável, o que facilita a manutenção e a introdução de novas funcionalidades conforme a solução cresce. A modularidade também assegura uma maior flexibilidade ao longo do tempo.
- **Base de Dados SQLite:** A utilização do SQLite [3] proporciona uma solução leve e eficiente para a gestão dos dados dos utilizadores e dispositivos. É uma escolha prática para protótipos e implementações iniciais devido à sua simplicidade e independência de um servidor dedicado.
- **Segurança e Criptografia:** A segurança é uma prioridade no desenvolvimento da solução, com a implementação de criptografia tanto na comunicação de dados quanto no armazenamento de informações sensíveis. Isso garante que as transações e os dados dos utilizadores estejam sempre protegidos.

5.2 Aplicação de Disciplinas do Curso

Este projeto é a convergência de várias disciplinas do curso de Engenharia Informática. As principais áreas de conhecimento aplicadas incluem:

- **Arquitetura de Software:** A solução foi projetada com uma arquitetura escalável e modular, adequada ao contexto de um sistema que precisa suportar vários utilizadores e dispositivos, como os condomínios.

- **Redes e Comunicação de Dados:** A integração entre as smart locks e a aplicação é realizada através de redes seguras, com a utilização de protocolos apropriados para garantir uma comunicação estável e protegida.
- **Desenvolvimento Web e Mobile:** A implementação de interfaces intuitivas e funcionais é feita com base nas tecnologias mais adequadas para o desenvolvimento de aplicações web e móveis, como o React para o front-end e Node.js para o back-end.
- **Segurança Informática:** A proteção dos dados é garantida por mecanismos de criptografia e por protocolos de segurança robustos, assegurando que as informações do utilizador e do sistema sejam sempre protegidas contra acessos não autorizados.

5.3 Requisitos Funcionais

Os requisitos funcionais descrevem as funcionalidades essenciais do sistema OP - Opening Doors, definidas para atender às necessidades de gestão de acessos em grandes comunidades residenciais. Estas funcionalidades garantem a segurança, usabilidade e escalabilidade do sistema, abrangendo desde a gestão de utilizadores até o controlo de acessos em tempo real. A Tabela 5.1, apresentada abaixo, detalha estes requisitos, indicando as prioridades de implementação (Must, Should, Could) e os respetivos níveis de esforço (Baixo, Médio, Alto) para cada funcionalidade.

5.3.1 Contexto adicional

Os requisitos funcionais foram definidos com base nas necessidades específicas do sistema OP - Opening Doors. Alguns dos requisitos necessitam de contexto adicional para esclarecer a sua importância:

- **Gestão de Acessos:** Refere-se à capacidade do sistema de gerir permissões e autorizações de acesso de forma eficiente, garantindo a segurança de portas em tempo real.
- **Definição de horários de acesso:** Permite configurar períodos específicos durante os quais determinados utilizadores podem aceder às portas, otimizando o controlo para diferentes cenários (por exemplo, horários de trabalho).
- **Bloqueio/desbloqueio remoto:** Essencial para situações em que um administrador precise de intervir rapidamente para garantir ou restringir o acesso a uma porta.
- **Registo de tentativas de acesso:** Garante o registo de todas as tentativas de acesso, tanto autorizadas quanto não autorizadas, promovendo a rastreabilidade e a auditoria.
- **Aplicação móvel:** Facilita o controlo do sistema por administradores ou utilizadores diretamente de um dispositivo móvel, aumentando a acessibilidade.
- **Acesso autorizado offline:** Permite que utilizadores previamente autorizados acessem às portas mesmo na ausência de uma conexão com a internet, garantindo a continuidade operacional.

5.3.2 Tabela de Requisitos Funcionais

Requisito Funcional	Have	Nível de Esforço
Gestão de Utilizadores	Must	Médio
Registo de utilizadores	Must	Baixo
Atualização e remoção de utilizadores	Must	Médio
Atribuição de permissões específicas	Must	Médio
Gestão de Acessos	Must	Alto
Controlo de acessos em tempo real	Must	Alto
Definição de horários de acesso	Should	Médio
Bloqueio/desbloqueio remoto	Must	Médio
Interface centralizada	Must	Alto
Personalização de acessos	Should	Médio
Autenticação segura	Must	Alto
Registo de tentativas de acesso	Must	Médio
Alertas para acessos não autorizados	Should	Médio
Relatórios de padrões de acesso	Could	Médio
Notificações para administradores	Could	Baixo
Aplicação móvel	Should	Alto
Integração com sistemas externos	Could	Alto
Conexão com dispositivos inteligentes	Could	Médio
Suporte a crescimento de utilizadores	Should	Médio
Funcionalidades adicionais	Could	Médio
Painel de gestão intuitivo	Must	Médio
Ferramentas de análise	Could	Médio
Acesso autorizado offline	Should	Alto
Sincronização de dados	Should	Médio
Simplicidade de instalação	Must	Baixo
Usabilidade intuitiva	Must	Baixo

Tabela 5.1: Tabela de Requisitos Funcionais do OP - Opening Doors

Capítulo 6

Calendário

6.1 Calendário

Através da aplicação ClickUp [9], organizámos e monitorizámos as tarefas do projeto desde a fase inicial de investigação até à execução final, garantindo o cumprimento rigoroso dos prazos estabelecidos. O desenvolvimento foi estruturado em diversas etapas, cada uma desempenhando um papel crucial para o sucesso do projeto OP - Opening Doors. Este planeamento detalhado, representado no cronograma Gantt ilustrado na Figura 6.1, permitiu uma gestão eficiente do tempo e das atividades. As etapas principais foram:

- **Relatório:** Incluiu as fases de investigação, definição de requisitos, pesquisa de soluções no mercado e definição da solução final. Esta fase serviu de base para compreender o contexto do problema e planear o desenvolvimento do sistema.
- **Montagem do Arduino:** Envolveu a escolha dos materiais necessários, o envio e a receção de componentes e a montagem do hardware. Esta etapa garantiu a preparação do sistema físico para integração com o software.
- **Criação do Servidor Flask:** Abrangeu a pesquisa de tecnologias, a implementação do backend, a realização de testes e a finalização do sistema. O servidor Flask foi essencial para a gestão da lógica do sistema e a comunicação com os dispositivos.
- **Protótipo:** Incluiu a escolha da paleta de cores e identidade visual, o planeamento das páginas da interface gráfica e a criação do modelo inicial. Esta etapa garantiu que o sistema oferecesse uma interface amigável e funcional para os utilizadores finais.

Cada uma dessas etapas foi cuidadosamente planeada e executada, assegurando o progresso contínuo do projeto e permitindo a integração harmoniosa entre as componentes físicas e digitais do sistema.

OP - Opening Doors 2



Figura 6.1: Diagrama de Gantt

Capítulo 7

Conclusão

O projeto OP - Opening Doors encontra-se numa fase inicial, onde já foram concluídas etapas fundamentais, como a investigação, a definição de requisitos e o planeamento. Estas fases iniciais estabeleceram uma base sólida para o desenvolvimento, garantindo que o trabalho segue uma direção clara e alinhada com os objetivos definidos. O progresso realizado até agora demonstra não apenas a viabilidade técnica do sistema, mas também a sua relevância para resolver problemas práticos de controlo de acessos em diferentes contextos.

- O projeto está bem estruturado e com um planeamento detalhado, o que proporciona uma perceção clara da realidade do trabalho e das suas exigências.
- Apesar de ser uma fase inicial, é visível o potencial impacto positivo desta solução no dia a dia das comunidades.

Com a complexidade inerente à integração entre hardware, software e segurança, os desafios enfrentados até ao momento serviram para validar as escolhas técnicas e identificar oportunidades de inovação. Este trabalho mostra-se promissor não apenas pela sua proposta atual, mas também pelas possibilidades futuras de expansão e evolução.

Possibilidades futuras:

- Integração com tecnologias mais avançadas de IoT, permitindo uma experiência ainda mais dinâmica e automatizada.
- Reforço de protocolos de segurança e autenticação, garantindo maior proteção e confiança no sistema.
- Adição de funcionalidades inteligentes, como notificações automáticas e relatórios detalhados, para melhorar a gestão e o controlo de acessos.

Em suma, o OP - Opening Doors reflete o esforço contínuo para desenvolver uma solução inovadora e prática, capaz de oferecer flexibilidade, segurança e escalabilidade. O trabalho está em progresso, mas o planeamento realizado até agora permite uma execução eficiente e alinhada com as expectativas. Este projeto tem não só o potencial de responder aos problemas atuais, mas também de se adaptar e crescer com as necessidades futuras, consolidando-se como uma solução de referência no controlo de acessos e na automação inteligente.

Bibliografia

- [1] João P. Matos-Carvalho. *The Lusófona L^AT_EX Template User's Manual*. Lusófona University. 2024. URL: <https://github.com/jpmcarvalho/UL-Thesis>.
- [2] Pallets Projects. *Flask Documentation*. 2024. URL: <https://flask.palletsprojects.com/> (acedido em 12/2024).
- [3] SQLite. *SQLite Documentation*. 2024. URL: <https://www.sqlite.org/docs.html> (acedido em 12/2024).
- [4] August Home. *August Wi-Fi Smart Lock: Secure and Convenient Access*. Accessed on 2023-09. 2023. URL: <https://august.com/products/august-wi-fi-smart-lock> (acedido em 09/2023).
- [5] Nuki Home Solutions. *Nuki Smart Lock: Your Key to a Smarter Life*. Accessed on 2023-09. 2023. URL: <https://nuki.io/en/smart-lock/> (acedido em 09/2023).
- [6] Microsoft. *IoT for Beginners*. 2024. URL: <https://learn.microsoft.com/en-us/azure/iot-fundamentals/> (acedido em 12/2024).
- [7] Meta (Facebook). *React - A JavaScript library for building user interfaces*. 2024. URL: <https://react.dev> (acedido em 12/2024).
- [8] OpenJS Foundation. *Node.js - JavaScript runtime documentation*. 2024. URL: <https://nodejs.org/en/docs/> (acedido em 12/2024).
- [9] ClickUp. *ClickUp Application*. 2024. URL: <https://clickup.com/> (acedido em 12/2024).