



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 | 27 / 06 / 2025

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

Resumo

Este projeto centra-se no desenvolvimento de um sistema web de controlo remoto de portas, utilizando tecnologias modernas. O backend foi desenvolvido com Flask [2] para a gestão de lógica e endpoints RESTful, complementado por SQLite [3] para a gestão eficiente de dados. A comunicação com o hardware é assegurada por um módulo ESP8266 [4], que interage diretamente com um Módulo Relé [5] para o controlo de fechaduras elétricas e um Micro Servo Tower Pro SG90 [6] para a atuação mecânica. 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 ao administrador registar e gerir utilizadores, monitorizar estados das portas em tempo real e controlar as suas ações de forma dinâmica, alternando entre os estados "aberta" e "fechada".

A aplicação foi concebida com uma arquitetura robusta, que inclui validação de dados e mecanismos de segurança para o acesso às operações. Durante o processo de desenvolvimento, foram adotadas implementações iterativas e realizados testes contínuos, nomeadamente com o Postman, garantindo a fiabilidade e o desempenho do sistema.

Este trabalho contribui significativamente com uma solução tecnológica inovadora para problemas práticos do quotidiano, destacando-se pela sua flexibilidade, fiabilidade e potencial para futuras integrações com sistemas IoT. Além disso, demonstra a eficácia de tecnologias acessíveis na resolução de desafios reais, promovendo a inovação e reforçando a segurança em ambientes automatizados.

Capítulo 2

Abstract

This project focuses on the development of a web-based remote door control system, leveraging modern technologies. The backend was developed using Flask [2] for logic management and RESTful endpoints, complemented by SQLite [3] for efficient data management. Hardware communication is ensured by an ESP8266 [4] module, which directly interacts with a Relay Module [5] for controlling electric locks and a Micro Servo Tower Pro SG90 [6] for mechanical actuation. The primary motivation for this work was to address the growing demand for practical and secure solutions for access automation in residential and corporate environments. The implemented system allows the administrator to register and manage users, monitor door states in real-time, and dynamically control their actions, switching between "open" and "closed" states.

The application was designed with a robust architecture, including data validation and security mechanisms for access to operations. Throughout the development process, iterative implementations were adopted, and continuous testing, particularly with Postman, was performed, ensuring the system's reliability and performance.

This work significantly contributes an innovative technological solution to practical everyday problems, distinguished by its flexibility, reliability, and potential for future integration with IoT systems. Furthermore, it demonstrates the effectiveness of accessible technologies in solving real-world challenges, promoting innovation and enhancing security in automated environments.

Conteúdo

1	Resumo	2
2	Abstract	3
	Conteúdo	4
	Lista de Figuras	6
	Lista de Tabelas	7
3	Introdução	8
	Introducao	8
3.1	Motivação	8
3.2	Identificação do Problema	8
3.2.1	Introdução	8
3.2.2	Benchmarking	8
3.2.3	Viabilidade e Pertinência	9
3.2.4	Solução Proposta	9
3.2.5	Calendário	9
4	Benchmarking	10
4.1	Identificação de Soluções no Mercado	10
4.2	Relação Front-end to Back-end	11
5	Viabilidade e Pertinência	13
5.1	Viabilidade	13
5.2	Pertinência	13
6	Solução Proposta	15
6.1	Tecnologias Utilizadas	15
6.2	Aplicação de Disciplinas do Curso	15
6.3	Requisitos Funcionais	16
6.3.1	Contexto adicional	16
6.3.2	Tabela de Requisitos Funcionais	16
7	Testes e Validação	19
7.1	Introdução	19
7.2	Metodologia de Testes	19
7.3	Tabelas de Testes	20
7.4	Resultados dos Testes	20
7.5	Considerações Finais	21
8	Calendário	22
8.1	Calendário	22
9	Conclusão	23

Lista de Figuras

4.1	Fluxograma Front-end to Back-end	12
8.1	Diagrama de Gantt	22

Lista de Tabelas

4.1	Tabela de Comparação (Parte 1)	10
4.2	Tabela de Comparação (Parte 2)	10
4.3	Tabela de Comparação (Parte 3)	11
6.1	Tabela de Requisitos Funcionais do OP - Opening Doors (Vista de Administrador) - Parte 1	17
6.2	Tabela de Requisitos Funcionais do OP - Opening Doors (Vista de Administrador) - Parte 2 (Continuação da Tabela 6.1)	18
7.1	Tabela de Testes do Sistema OP - Opening Doors (Relé, Servo e API via Wi-Fi)	20

Capítulo 3

Introdução

3.1 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.

3.2 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.

3.2.1 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.

3.2.2 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.

3.2.3 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.

3.2.4 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.

3.2.5 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 4

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 [7], Nuki Smart Lock [8], 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.

4.1 Identificação de Soluções no Mercado

Abaixo estão as tabelas(4.1 , 4.2 , 4.3) 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 4.1: 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 4.2: 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 4.3: Tabela de Comparação (Parte 3)

4.2 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 utilizador é o ponto de partida, interagindo com o sistema através de uma interface gráfica acessível e responsiva, desenvolvida no frontend (possivelmente com React [9]).
- 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 [10].
- O backend, implementado com ferramentas como Flask [2], 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 [3], 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(4.1) fornece uma visão clara desse fluxo de interação, permitindo compreender como as diferentes partes se interligam para formar um sistema coeso e eficiente.

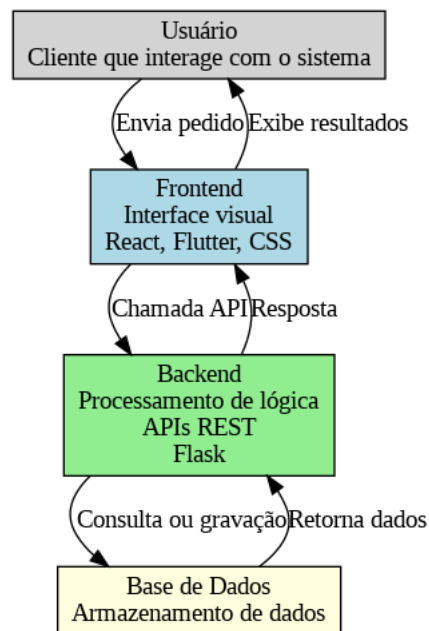


Figura 4.1: Fluxograma Front-end to Back-end

Capítulo 5

Viabilidade e Pertinência

5.1 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 [7] e a Nuki Smart Lock [8]. 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.

5.2 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 [7] e a Nuki Smart Lock [8], 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 6

Solução Proposta

O projeto OP - Opening Doors visa criar uma solução de smart lock funcional e acessível, especialmente concebida para utilização em condomínios e ambientes com múltiplos utilizadores. A proposta consiste em oferecer um sistema seguro e eficiente para controlo remoto de acessos, através de uma API REST desenvolvida em *Flask*[2], que comunica com dispositivos físicos *ESP8266*[4] via Wi-Fi. Estes dispositivos controlam um *Relay Module*[5], responsável por permitir ou interromper a alimentação de corrente a um atuador, como um *Micro Servo Tower Pro SG90*[6], simulando assim a abertura de portas. A plataforma foi desenvolvida de forma modular, permitindo a gestão de utilizadores, portas e logs de acesso, garantindo escalabilidade e facilidade de integração futura com novas funcionalidades ou dispositivos.

6.1 Tecnologias Utilizadas

- **Internet das Coisas (IoT) e Smart Locks:** A utilização de dispositivos IoT [11] 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.
- **API REST com Flask:** A plataforma implementada utiliza um servidor *Flask* [2], responsável por expor uma API REST para controlo e gestão de portas, utilizadores e logs de acessos.
- **Comunicação Wi-Fi com ESP8266:** A comunicação com os dispositivos físicos é realizada via Wi-Fi, utilizando módulos *ESP8266* [4], que recebem os comandos da API e acionam os periféricos.
- **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.

6.2 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:** 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.
- **Computação Móvel:** 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.

6.3 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 ??, 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.

6.3.1 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 e Logs:** Refere-se à capacidade do sistema de gerir permissões e autorizações de acesso de forma eficiente, garantindo a segurança de portas em tempo real. Inclui não só o controlo remoto, mas também o registo de logs com data e hora de acionamento de cada porta.
- **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:** Abertura e fecho de portas realizadas remotamente via comandos HTTP para o ESP8266.
- **Ativação de Relé e Servo:** Comando remoto para alternar o estado do *Relay Module* e movimentação do *Micro Servo* em tempo real.

6.3.2 Tabela de Requisitos Funcionais

Requisito Funcional	Have	Nível de Esforço	Descrição
Gestão de Utilizadores	Must	Médio	O sistema deve permitir que o administrador crie, visualize, edite e remova contas de utilizadores que interagem com o sistema de acesso/portas. Cada utilizador deve ter um nome, palavra-passe segura
Registo de utilizadores	Must	Baixo	O sistema deve permitir ao administrador registar novos utilizadores, atribuindo-lhes credenciais iniciais e controlando as suas permissões de acesso.
Atualização e remoção de utilizadores	Must	Médio	O sistema deve permitir ao administrador atualizar informações de perfis de utilizador existentes, como permissões, e remover contas de utilizadores conforme necessário.
Atribuição de permissões específicas	Must	Médio	O sistema deve permitir ao administrador atribuir permissões de acesso granulares a diferentes funcionalidades do sistema ou a portas específicas para cada utilizador.
Gestão de Acessos	Must	Alto	O sistema deve permitir ao administrador configurar e supervisionar as regras e políticas que governam o acesso a diferentes portas e funcionalidades do sistema.
Controlo de acessos em tempo real	Must	Alto	O sistema deve permitir ao administrador monitorizar e controlar o estado das portas (abrir/fechar) em tempo real, recebendo feedback imediato sobre as operações.
Bloqueio/desbloqueio remoto	Must	Médio	O sistema deve permitir ao administrador bloquear ou desbloquear remotamente portas individuais ou grupos de portas através da interface de gestão.
Interface centralizada	Must	Alto	O sistema deve providenciar uma interface de utilizador web/móvel centralizada para que o administrador possa gerir todas as funcionalidades do sistema (utilizadores, portas, registos).
Autenticação segura	Must	Alto	O sistema deve implementar mecanismos robustos de autenticação (ex: palavras-passe criptografadas, gestão de sessões) para proteger o acesso do administrador e dos utilizadores ao sistema.

Tabela 6.1: Tabela de Requisitos Funcionais do OP - Opening Doors (Vista de Administrador) - Parte 1

Requisito Funcional	Have	Nível de Esforço	Descrição
Integração com sistemas externos	Could	Alto	O sistema poderá integrar-se com outros sistemas (ex: sistemas de segurança, gestão de edifícios) para interoperabilidade e funcionalidades expandidas.
Conexão com dispositivos inteligentes	Could	Médio	O sistema poderá estabelecer e manter uma conexão robusta com os dispositivos de smart lock (ex: ESP8266) para controlo e monitorização contínuos.
Suporte a crescimento de utilizadores	Should	Médio	O sistema deve ser escalável para suportar um número crescente de utilizadores e portas sem comprometer o desempenho ou a estabilidade.
Funcionalidades adicionais	Could	Médio	O sistema poderá incorporar funcionalidades avançadas (ex: reconhecimento facial, integração com assistentes de voz) para melhorar a experiência e segurança.
Painel de gestão intuitivo	Must	Médio	O sistema deve apresentar um painel de gestão para o administrador que seja fácil de usar e navegar, com informações claras e acesso rápido às funcionalidades principais.
Ferramentas de análise	Could	Médio	O sistema poderá oferecer ferramentas que permitam ao administrador analisar dados de acesso e comportamento para identificar tendências ou anomalias.
Sincronização de dados	Should	Médio	O sistema deve garantir a sincronização consistente dos dados entre o servidor central e os dispositivos de acesso, assegurando que as informações de permissão estejam sempre atualizadas.

Tabela 6.2: Tabela de Requisitos Funcionais do OP - Opening Doors (Vista de Administrador) - Parte 2 (Continuação da Tabela 6.1)

Capítulo 7

Testes e Validação

7.1 Introdução

A fase de testes e validação desempenha um papel essencial no desenvolvimento de soluções tecnológicas, sendo determinante para garantir a qualidade, robustez e fiabilidade do sistema implementado. No projeto *OP - Opening Doors*, esta fase permitiu assegurar que todas as funcionalidades previstas foram corretamente implementadas e que a comunicação entre o servidor *Flask* [2], a base de dados *SQLite* [3] e os dispositivos físicos *ESP8266* [4], *Relay Module* [5] e *Micro Servo SG90* [6] ocorre de forma estável e sem falhas.

Os testes foram estruturados para validar individualmente os principais módulos do sistema, com especial foco na interação entre utilizadores, API REST, controlo remoto de portas, e comunicação via Wi-Fi com o Arduino e os seus periféricos. Esta fase incluiu ainda a verificação do correto funcionamento do Relay Module para controlo de alimentação e do servo motor para simulação de abertura física da porta.

7.2 Metodologia de Testes

Para garantir a fiabilidade da aplicação desenvolvida, definiu-se uma metodologia de testes composta por testes funcionais, testes de integração e testes de comunicação física. Os testes funcionais validaram o comportamento da API, enquanto os testes de integração asseguraram a comunicação entre back-end, base de dados e os dispositivos.

A comunicação com o *ESP8266* passou a ser exclusivamente via Wi-Fi, eliminando a utilização da comunicação serial USB. As requisições foram simuladas através do *Postman* [10], enviando comandos HTTP diretamente para o microcontrolador, que controla o Relay Module e o Micro Servo SG90.

Configuração utilizada:

- Computador com Windows 11.
- ESP8266 [4] ligado via cabo USB apenas para alimentação.
- Relay Module [5] conectado ao pino D2 (GPIO4).
- Micro Servo Tower Pro SG90 [6] no pino D5 (GPIO14).
- Visual Studio Code.
- Arduino IDE 2.3.2 [12].
- Postman [10].
- Flask 2.3.2 [2].
- Base de dados SQLite 3.45.1 [3].

7.3 Tabelas de Testes

No [vídeo](#) é possível observar todos os testes da tabela abaixo 7.1

Título	Descrição	Resultado Esperado
Registo de Utilizador	Enviar pedido POST para o endpoint <code>/register</code> com <i>username</i> e <i>password</i> .	Utilizador criado e registado na base de dados; código HTTP 201 e resposta JSON com a mensagem de sucesso.
Criação de Porta	Enviar POST para <code>/create-door</code> com <i>name</i> , <i>user_id</i> , <i>arduino_ip</i> .	Porta registada na base de dados e associada ao utilizador; código HTTP 201 e resposta JSON confirmando a criação da porta.
Consulta de Portas	Enviar GET para <code>/doors</code> .	Listagem de todas as portas registadas; código HTTP 200.
Alternar Estado da Porta (Wi-Fi)	Enviar POST para <code>/toggle-door</code> com o <i>door_id</i> .	Estado alternado e comando enviado via Wi-Fi para o Arduino; código HTTP 200.
Controlo do Relay Module	Enviar comando <code>/toggle-door</code> via API.	Relé ativa e desativa corretamente, confirmável pela carga ou LED; código HTTP 200.
Controlo do Micro Servo	Enviar comando via API que mova o servo através do Arduino.	Micro servo move-se para as posições definidas; código HTTP 200.
Log de Abertura de Porta	Consultar base de dados após toggle.	Registo de data/hora atualizado no campo <code>last_opened_at</code> na base de dados.

Tabela 7.1: Tabela de Testes do Sistema OP - Opening Doors (Relé, Servo e API via Wi-Fi)

7.4 Resultados dos Testes

Os testes confirmaram o correto funcionamento da API e das comunicações Wi-Fi. Todos os pedidos realizados via Postman [10] resultaram na ativação e desativação do relé [5], e no movimento do micro servo [6], de acordo com os comandos enviados. O Relay Module [5] respondeu de forma imediata aos pedidos HTTP, e o micro servo executou movimentos suaves e precisos.

Foi registada uma ligeira latência inferior a 500ms em comandos via Wi-Fi, aceitável para este tipo de aplicação doméstica.

A verificação de logs na base de dados permitiu confirmar a atualização correta dos estados e timestamps de cada porta acionada.

Adicionalmente, foram realizados testes repetitivos, enviando mais de 50 comandos consecutivos via *Postman* [10], sem registo de falhas de comunicação ou inconsistências

no comportamento dos dispositivos. Testes com dois dispositivos *ESP8266* configurados em simultâneo permitiram validar a escalabilidade e a estabilidade do sistema, confirmando a capacidade da aplicação para gerir múltiplas portas e comandos concorrentes em rede local.

7.5 Considerações Finais

A fase de testes e validação foi crucial para garantir a fiabilidade do sistema *OP - Opening Doors*. A integração entre API, Relay Module [5], Micro Servo [6] e comunicação Wi-Fi foi bem-sucedida e robusta, permitindo a expansão futura para mais dispositivos.

O projeto encontra-se funcional e validado, estando preparado para integração de controlo físico de portas reais e expansão futura com múltiplos dispositivos e sensores conectados.

Capítulo 8

Calendário

8.1 Calendário

Através da aplicação ClickUp [13], 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 8.1, 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.

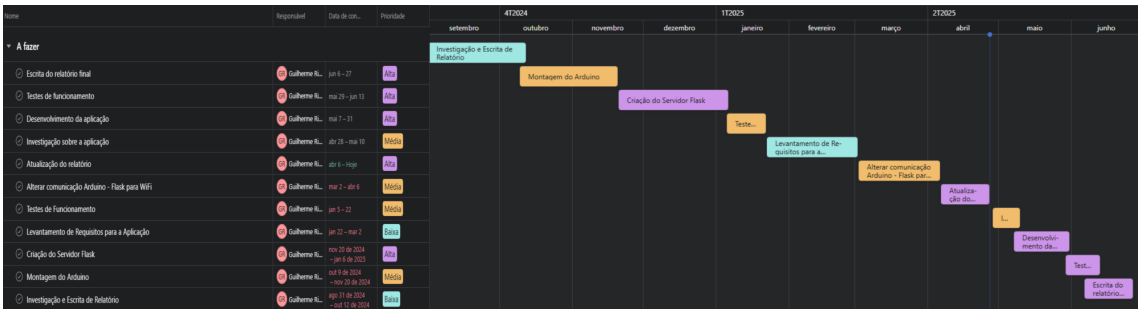


Figura 8.1: Diagrama de Gantt

Capítulo 9

Conclusão

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

- O projeto está bem estruturado e com um planeamento detalhado, o que proporciona uma perceção realista das exigências técnicas e dos recursos necessários para a sua execução.
- Apesar de se tratar de uma fase inicial, é evidente o potencial impacto positivo desta solução no quotidiano das comunidades e na gestão inteligente de acessos.

Com a complexidade inerente à integração entre hardware, software e segurança informática, os desafios enfrentados até ao momento serviram para validar as escolhas tecnológicas e identificar oportunidades concretas de inovação. Este trabalho evidencia-se não apenas pela proposta atual, mas também pelo seu potencial de crescimento e adaptação.

Possibilidades futuras:

- Integração com tecnologias mais avançadas de IoT, incluindo sensores ambientais e câmaras IP, proporcionando uma experiência ainda mais automatizada e interativa.
- Desenvolvimento de uma aplicação móvel dedicada, permitindo aos utilizadores finais e administradores efetuar a gestão e o controlo de acessos remotamente de forma prática e segura, com notificações em tempo real e histórico de acessos.
- Reforço de protocolos de segurança e autenticação, implementando soluções como tokens de sessão, OAuth e encriptação ponta-a-ponta nas comunicações.
- Adição de funcionalidades inteligentes, como notificações automáticas, relatórios detalhados de acessos e integração com assistentes de voz e plataformas de domótica.

Em suma, o *OP - Opening Doors* reflete o esforço contínuo para desenvolver uma solução inovadora, prática e segura, capaz de oferecer flexibilidade, escalabilidade e integração com diferentes ambientes. O trabalho encontra-se numa fase promissora, com o planeamento e a execução até aqui assegurados de forma eficiente e alinhada com as expectativas. Este projeto apresenta-se como uma proposta viável e adaptável às necessidades futuras, consolidando-se como uma solução de referência no domínio da gestão inteligente de acessos e automação de espaços.

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] Espressif Systems. *ESP8266EX Datasheet and Technical Documentation*. Accessed on 2024-12. 2024. URL: <https://www.espressif.com/en/products/socs/esp8266>.
- [5] Keyes DIY. *Relay Module Guide — How to Use Relay Modules with Arduino*. Accessed on 2024-12. 2024. URL: <https://lastminuteengineers.com/relay-module-interfacing-arduino/> (acedido em 12/2024).
- [6] Tower Pro. *Tower Pro SG90 Micro Servo — Datasheet and Guide*. Accessed on 2024-12. 2024. URL: <https://towerpro.com.tw/product/sg90-9g-micro-servo/> (acedido em 12/2024).
- [7] 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).
- [8] 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).
- [9] Meta (Facebook). *React - A JavaScript library for building user interfaces*. 2024. URL: <https://react.dev> (acedido em 12/2024).
- [10] Postman. *Postman Documentation*. 2024. URL: <https://www.postman.com/> (acedido em 12/2024).
- [11] Microsoft. *IoT for Beginners*. 2024. URL: <https://learn.microsoft.com/en-us/azure/iot-fundamentals/> (acedido em 12/2024).
- [12] Arduino. *Arduino IDE Documentation*. Accessed on 2024-12. 2024. URL: <https://docs.arduino.cc/software/ide-v2>.
- [13] ClickUp. *ClickUp Application*. 2024. URL: <https://clickup.com/> (acedido em 12/2024).