UNIVERSIDADE
LUSÓFONA

Sistemas de abertura de portas das salas de aulas

Trabalho Final de curso

Relatório Intercalar 2º Semestre

Nome do Aluno: João Joyce

Nome do Aluno: Tiago James

Nome do Orientador: José Brás

Nome do Co-orientador: Luís Gomes

Trabalho Final de Curso | LEI | 28 Jun 2024

www.ulusofona.pt

Direitos de cópia

João Joyce, Tiago James, Copyright de João Joyce, Tiago James, 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.

Resumo

O projeto Sistemas de Abertura de Portas (TFC DEISI 12) visa melhorar o funcionamento diário dos docentes e alunos da Universidade Lusófona de Humanidades e Tecnologia, substituindo o atual sistema de requisição de chaves. A proposta, apresentada por docentes da universidade, envolve a implementação de um sistema de abertura de portas sem a intervenção do PUI.

O projeto começa com a descrição do problema atual, destacando as desvantagens do sistema existente. Em seguida, um benchmarking compara a nova solução com sistemas de mercado, como os usados em hotéis. A viabilidade e pertinência mostram como a solução pode trazer mais eficiência à universidade, com potencial para expansões e melhorias futuras.

Os requisitos, diagramas e estruturas de suporte ao projeto são apresentados e utilizados como guia na fase de implementação. A proposta inclui a indicação das fases do projeto, o hardware necessário, o software a ser desenvolvido e o funcionamento do sistema. O sistema consiste em uma fechadura que utiliza um cartão RFID para garantir ou negar acesso às salas.

Sendo este o relatório final todas as fases de desenvolvimento foram concluídas e o sistema testado, foi também terminada a maquete final que demonstra o sistema em uso.

Abstract

The Door Opening Systems project (TFC DEISI 12) aims to improve the daily functioning of teachers and students at the Lusófona University of Humanities and Technology by replacing the current key request system. The proposal, presented by university teachers, involves implementing a door opening system without the need for PUI intervention.

The project begins with a description of the current problem, highlighting the disadvantages of the existing system. Then, a benchmarking compares the new solution with market systems, such as those used in hotels. The viability and relevance show how the proposed solution can bring more efficiency to the university, with potential for future expansions and improvements.

The project requirements, diagrams, and support structures are presented and used as a guide during the implementation phase. The proposal includes the project phases, the necessary hardware, the software to be developed, and the system's operation. The system consists of a lock that uses an RFID card to grant or deny access to the rooms.

As this is the final report, all development phases have been completed and the system has been tested. The final model demonstrating the system in use has also been finished.

Índice

Resumo iii
Abstract iv
Índice v
Lista de Figuras vii
Lista de Tabelas viii
1. Identificação do Problema 1
2. Benchmarking 2
3. Viabilidade e Pertinência 4
Pertinência 5
Viabilidade 5
4. Engenheira 7
4.1. Levantamento e análise dos Requisitos 7
4.1. Diagramas de Casos de Uso 9
4.2. Diagramas de sequência e BPMN 10
4.3. Modelos relevantes 12
4.4. Estrutura 13
4.5. Mock ups 14
4.6. Modelo 3D 16
5. Solução Proposta 17
5.1. Introdução 17
5.2. Arquitetura 17
5.3. Tecnologias e ferramentas utilizadas 18
5.4. Implementação 19
5.5. Abrangência 20
6. Plano de testes e avaliação 21
7. Método e Planeamento 22
8. Resultados 24
9. Conclusão e Trabalhos Futuro 25
Bibliografia 26
Anexo 1 - Questionário 27
Anexo 2 - Código SQL 31

Anexo 3 - Calendário 35
Glossário 38

Lista de Figuras

Figura 1. - Diagrama de Casos de Uso 9

Figura 2. - Diagrama de Sequência - usar o sistema 10

Figura 3. - Diagrama - Processo de inscrição 10

Figura 4. - Diagrama BPMN As-Is 11

Figura 5. - Diagrama BPMN To-Be 11

Figura 6. - Modelo da Base de Dados 12

Figura 7. - Diagrama de Arquitetura do Sistema 13

Figura 8. - Login 14

Figura 9. - Página principal 15

Figura 11. - Formulário de registo de utilizador 15

Figura 12. - Modelo 3D 16

Figura 13. - Diagrama do sistema 17

Figura 14. - Fechadura solenoide 18

Figura 15. - Raspberry Pi 0 19

Figura 16. - Leitor RFID RC522 e Cartão 19

Figura 17. - Esquema de ligação Raspberry Local 20

Figura 18. - Calendário Gantt 22

Figura 19. - Calendário Gantt 37

Lista de Tabelas

Tabela 1. - Tabela de custos 2
Tabela 2. - Requisitos funcionais 8
Tabela 3. - Requisitos não funcionais 9
Tabela 4. - dados “access_level” 12
tabela 5. - Calendário 22
Tabela 6. - Código de cores 35
Tabela 7. - Calendário.....36

1. Identificação do Problema

Na faculdade está em funcionamento o Ponto único de informação (PUI), cuja uma das suas funções é a recolha e distribuição de chaves a professores para as salas da faculdade. Porém com as inovações tecnológicas este sistema pode ser inovado. Com a necessidade de usar uma chave os docentes são forcados a entregar e recolher uma chave no PUI para cada sala, o que pode ser menos cómodo quando a sala em questão é uma sala no mesmo bloco ou adjacente. Este sistema origina uma perda de tempo, pois adiciona mais uma "tarefa" entre aulas. Outras problemáticas a identificar com este sistema são a manutenção das chaves e do PUI, isto engloba a perda de chaves ou danificação das mesmas, bem como o gasto salarial em funcionários para manter o espaço aberto e funcional sempre que a faculdade também se encontra em funcionamento.

A solução desenvolvida neste projeto visa a atualização do sistema de chaves para um sistema de abertura de portas com uso do cartão emitido pela faculdade, não só este sistema trás facilidade e conveniência aos docentes como também possivelmente aos alunos, que através da funcionalidade de requisição de salas junto da coordenação dos departamentos, podem tomar partido desta solução para requisitar a sala e não tendo necessidade de recolher uma chave mas sim utilizar o cartão de estudante emitido pela universidade.

Através do questionário apresentado no Anexo 1. aplicado foi possível concluir que apesar do sistema atual não ser fonte de uma problemática para o dia-a-dia para os professores, nenhum docente se opõe a atualização do mesmo. Existindo alguns docentes que partilharam quais funcionalidades um sistema deste tipo poderia implementar.

Face ao problema identificado, com a implementação do sistema os nossos resultados indicam que é criada uma comodidade e conveniência adicional, não sacrificando em simplicidade.

2. Benchmarking

Existem sistemas com integração menos profunda daquele a que se propõe a solução desenvolvida neste trabalho, como por exemplo na entrada para empresas as portas de entrada também usam um sistema com recurso a chaves RFID, neste caso o sistema é utilizado principalmente por motivos de segurança, bloqueando entrada a terceiros. Estes sistemas integram num Back office que gere os acessos.

Podem ser encontrados sistemas similares em hotéis, apesar de com algumas diferenças. Nomeadamente, nas chave em si, (em hotéis é usada uma chave genérica onde é gravado o acesso ao quarto) nesta solução a chave é pessoal (Cartão de docente/aluno), o que implica a implementação de um um Back-office onde é feita a gestão de acessos, esta implementação significa que o sistema pode ser alterado dinamicamente consoante as necessidades da faculdade, sendo que com o desenrolar desta solução o sistema de gestão de acessos pode ser refinado para ter uma apresentação mais apelativa para o utilizador, levando á facilidade de uso da mesma.

Comparativamente com sistemas utilizados em hotéis - onde todo o hardware está contido na fechadura em si - a solução apresentada neste trabalho é modular, ou seja, os componentes usados não são integrados num único equipamento, apesar de ser necessário uso de espaço adicional para a implementação, também significa que todos os componente podem ser alterados. Havendo esta modularidade o sistema apresenta uma reparabilidade maior e uma maior facilidade de melhoria dos componentes, ou seja, é possível a troca e adição de componentes por iguais ou melhores.

Este projeto foi desenvolvido com o âmbito de ser aplicado a todas as portas da universidade lusófona, por isso, devido á escala a que se propõe, a solução proposta tem como um dos seus objetivos ser o mais económica possível sem comprometer as funcionalidades a que se propõe. A escolha dos componentes a serem usados é feita com base no seu custo médio, o objetivo é obter uma solução consciente de custo onde não se sacrifica a qualidade das funcionalidades.

Na Tabela 1. é apresentado uma analise de custos sendo evidenciado o custo do desenvolvimento do projeto e o custo de recursos humanos para operar o PUI.

	Custos de desenvolvimento do projeto:		**Potenciais custos poupados**
Software:			
Desenolvimento do código realizado durante o TFC	€0		
Hardware:			
Raspberry Pi:	€18,30		
Sensor RFID (+40) :	€2,03 (cada)		
LEDs - Vermelho (10):	€0,10 (cada)		
LEDs - Verde (+10):	€0,10 (cada)		
Fechadura (+10):	€6,70 (Cada)		

	Custos de desenvolvimento do projeto:		**Potenciais custos poupados**
Total (Por porta):	**€27,23**		
		Custo operacional anual (1 pessoa):	€19 301,7
Total (p/ 40 portas)	**€601,2**	**Total (1 pessoa e meia):**	**€28 952,55**

Tabela 1. - Tabela de custos[1]

Para calcular o custo para um dado numero de portas é utilizada a seguinte formula :

$$\left(2.03 + 0.10 + 0.10 + 6.70\right) \times x + \left(18.30 \times \left(\frac{x}{3}\right)\right)$$

Onde x representa o numero de portas desejado.

Nesta tabela não é equacionado o custo de cabelagem, uso de energia ou instalação. É de de notar também que uma fechadura que seja comercializada com todos estes componentes integrados tem o valor medio de ~150€, dependendo das funcionalidades especificas a cada sistema.

[1] Valores do Hardware retirados do website: mauser.pt

3. Viabilidade e Pertinência

A solução proposta foi desenvolvida para ser aplicada em grande escala, e por isso, existe uma atenção especial relativa ao custo de cada unidade de abertura automática. Querendo ter disponíveis as funcionalidades mais úteis e práticas para os utilizadores, um sistema base deve ser apresentado como uma fechadura solenóide com pelo menos dois tipos de acesso, um eletrónico e um mecânico ("Fail Safe"), em caso de falha no fecho eletrónico. Um leitor RFID, pois o chip RFID está presente em quase todos os cartões emitidos atualmente, reduzindo os custos relacionados á criação e distribuição de chaves de acesso. Comparativamente a outros sensores o par leitor RFID e cartão apresenta um custo de mercado inferior. A gestão de acessos será feita com recurso a programação de unidades de Raspberry Pi, onde o objetivo é a redução do numero de Raspberry Pi por cada porta, ou seja, não ter um Raspberry Pi por cada uma, maximizando as capacidades de cada Raspberry Pi instalado. O Sistema vai ser conectado através de um router em modo de Access Point, permitindo assim a comunicação entre o controlador e o servidor.
Porém o sistema não deixa de ser expansível, melhorias podem ser aplicadas futuramente, como sensores biométricos, levando á eliminação total de chave física, fechaduras totalmente electrónicas. Com recurso a um UPS como Failsafe , será possível ter um plano de emergência no caso de falha energética. Numa possível analise a custos o uso de Raspberry Pi com capacidades de output e input superiores podem ser úteis em reduzir o numero de unidades Raspberry Pi, sendo que estes poderão controlar mais sensores por cada unidade de Raspberry Pi.

É expectável que com a eliminação da funcionalidade oferecida pelo PUI haja uma fase de adaptação mas que as vantagens de um sistema como o descrito a cima serão rapidamente notadas. Os recursos humanos alocados ao PUI podem ser alocados a outra tarefa mais importante/critica tendo eles também um maior envolvimento no bom funcionamento da instituição, tanto como nos docentes que vão experienciar uma fluidez/eficiência de funcionamento superior. Foi realizado um inquérito aos docentes, onde se procura saber a apreciação feita pelos docentes do serviço prestado pelo PUI e sobre o sistema a ser desenvolvido neste projeto.

Através da analise feita com base no inquérito realizado (resultados apresentados no anexo 1.), podemos concluir que apesar do sistema atual não ter um funcionamento debilitante, nenhum docente estaria oposto á implementação de um novo sistema, que toma partido das tecnologias atuais.

Com base nos dados recolhidos, compreendido entre uma escala de 0 a 5, respostas sim ou não e resposta aberta, a primeira questão referia-se à avaliação do processo de requisição e entrega de chaves em termos de rapidez, a maioria dos docentes avaliou o processo com uma pontuação de 4 , o que indica uma avaliação positiva quanto à sua rapidez. A segunda questão tinha como objetivo compreender a frequência com que os docentes são expostos a um tempo de espera elevado quando se trata de obter chaves do PUI, neste caso a resposta variou entre nada frequente (1) a relativamente frequente (4), o facto de termos tido o mesmo número de votos não nos permite aferir com certezas esta questão, podendo apenas concluir que é um acontecimento que pode acontecer ocasionalmente. Ainda sobre tempo de espera, obtivemos um resultado de maioria (55.5% dos docentes) no que trata a se alguma vez foi necessário esperar que a chave que necessita fosse entregue no PUI, estes afirmaram que sim. Já sobre a dificuldade na abertura das portas 66.7% admite não sentir essa dificuldade com frequência.

Também com base nas respostas fornecidas pelos docentes no questionário anónimo, a maioria expressou que não considerava extremamente inconveniente ter que se deslocar ao PUI sempre que necessitavam de trocar de sala, havendo, porem, mais de 75% dos docentes a responder entre os valores 2 a 4 . Quando questionados sobre sua satisfação geral com os serviços prestados pelo PUI, os docentes indicaram que se sentem bastante satisfeitos, mas não totalmente, o que sugere uma avaliação positiva, mas com espaço para desenvolvimento. Adicionalmente, todos os docentes indicaram que, se existisse um sistema que permitisse a não deslocação ao PUI para requisição e entrega de chaves, estariam interessados em adotar esse sistema. Este resultado enfatiza a necessidade percebida de uma solução alternativa que facilite o acesso às salas de aula, a maioria dos docentes expressou, também, que seria vantajoso ter um ecrã informativo em cada sala, contendo informações relevantes como o nome do docente e a hora de ocupação. Isso sugere uma demanda por maior transparência e conveniência no ambiente académico. Além disso, os docentes manifestaram preferência pelo uso de um PIN pessoal de acesso em vez do cartão de docente da faculdade. Isso indica uma inclinação para métodos de autenticação mais personalizados e possivelmente mais seguros no processo de acesso às salas de aula. Por último, conseguimos também entender que os docentes confiariam num sistema biométrico para acesso ás salas de aulas.

Pertinência

No âmbito de transformar o acesso aos espaços de ensino dos docentes e alunos da Universidade Lusófona de Humanidades e Tecnologia, este projeto visa assegurar o acesso a salas do estabelecimento recorrendo a um sistema de abertura de portas automático de forma a facilitar o processo atual sem que os mesmos tenham de recorrer ao PUI, excepto em ultimo caso.

Com o sistema de portas pretende-se que os acessos às salas sejam disponibilizados através do uso de cartões que serão apresentados num leitor junto da sala a que se pretende ter acesso, posteriormente o sistema irá verificar o devido acesso, garantindo, ou não, o acesso à sala.

Viabilidade

O Sistema de portas é direcionado aos docentes e alunos para que estes possam usufruir dos espaços da faculdade, ainda que anteriormente reservados mediante disponibilidade de horários ou reservas junto da coordenação. Em suma, o desenvolvimento deste projeto irá facilitar a poupança dos recursos despendidos, o tempo de espera, a aquisição de chaves e também os deslocamentos. A continuidade ou replicação desta solução para mais departamentos, faculdade inteira ou até mesmo novos espaços ou outras instalações pertencentes à faculdade será possível. Adicionalmente, este projeto tem potencial para agregar mais componentes futuros, como, por exemplo, uma sugestão de um docente da faculdade sobre a utilidade que poderia ser obtida em servir de auxílio para o registo de presenças.

Proposto inicialmente foi a implementação de um sistema de fechadura com RFID e acesso mecânico, o uso de Raspberry Pi para a gestão de acessos, comunicação via router em modo de *Access Point* e a expansibilidade futura com sensores biométricos. O que foi desenvolvido incluiu a implementação completa do sistema de fechadura com RFID e acesso mecânico, gestão de acessos centralizada com Raspberry Pi4 e comunicação eficiente entre controlador e servidor via Access Point.

Durante o desenvolvimento, optou-se por utilizar o Flask, uma ferramenta do Python, para implementar um Back-office acessível através de uma página web. Esta decisão optimizou significativamente o processo de desenvolvimento, facilitando a gestão e monitorização do sistema de forma mais intuitiva e acessível aos administradores.

As alterações introduzidas durante o desenvolvimento não comprometeram a viabilidade nem a pertinência do projeto. Pelo contrário, as otimizações feitas na gestão de Raspberry Pi e na comunicação entre os componentes reforçaram a eficiência do sistema. A possibilidade de expansão futura com sensores biométricos e outras funcionalidades demonstra a flexibilidade e o potencial de crescimento do projeto.

Os resultados dos testes de validação confirmam que a solução desenvolvida atende aos requisitos iniciais e vai além, oferecendo uma plataforma robusta e expansível que pode ser adaptada a necessidades futuras. Assim, o projeto mantém-se viável e pertinente, cumprindo os objetivos propostos e garantindo benefícios significativos para os utilizadores finais.

4. Engenheira

4.1.Levantamento e análise dos Requisitos

Para os requisitos do sistema foi desenvolvida uma tabela MoSCoW onde definimos os requisitos como sendo “Must Have”;”Should Have”;”Could Have” e “Won’t Have”, onde “Must Have” são os requisitos que o sistema deve implementar para o funcionamento base proposto, “Should Have” os requisitos que o sistema devia ter porem não são críticos para o seu funcionamento, "Could Have” são os requisitos que poderiam ser implementados mas que para esta proposta consideramos como sendo apenas embelezamento e que não respondem diretamente ao problema identificado e “Won’t Have” requisitos que classificamos como possíveis extensões da solução proposta mas não serão desenvolvidas no âmbito deste trabalho. Na ultima coluna é representado quais os requisitos implementados, sendo “I” *Implemented*, ”PI” *partially Implemented* e ”NI” *Not Implemented*. O requisito “PI” FR-15 difere no acesso departamental, no sistema final qualquer cartão de docente abre qualquer porta, no requisito NFR-7 a parte não implementada é o Back-office que não tem inclui a funcionalidade de criar novos utilizadores para efetuar login, esse registo tem de ser feito na BD.

ID	Description	MoSCoW	As is/To be	Classification	Dev. Effort	I/NI
FR-1	O Sistema deve permitir o registo cartões	Must Have	To improve	Configuration	M	I
FR-2	O sistema deve permitir a alteração de dados associados a cartões	Must Have	To improve	Configuration	M	I
FR-3	O sistema deve permitir a remoção de cartões do mesmo	Must Have	To improve	Configuration	M	I
FR-4	Quando apresentado um cartão com acesso total no sensor RFID é aberta a porta da sala	Must Have	To improve	Bespoke development	M	I
FR-5	Um cartão com acesso departmental pode apenas abrir portas do departamento associado ao mesmo	Should Have	To improve	Bespoke development	M	NI
FR-6	Um aluno (ou outro) apenas pode abrir uma sala se tiver requisitado a mesma	Could Have	To improve	Configuration	M	I
FR-7	Quando garantido acesso a sala um LED verde acende	Should Have	To improve	Out of the box	M	I
FR-8	Quando não garantido acesso a sala um LED vermelho acende	Should Have	To improve	Out of the box	M	I
FR-9	Só é garantido acesso a uma sala a um doecente se for a sala registada no seu horário	Could Have	To improve	Bespoke development	M	NI
FR-10	O acesso as sala pode ser feito por meios mecânicos, em caso de falha energética	Should Have	To improve	Out of the box	M	I
FR-11	O sistema deve ter um UPS	Could Have	To improve	Configuration	M	I
FR-12	Deverá ser implementado um conjunto de diferentes níveis de acesso	Could Have	To improve	Bespoke development	M	I
FR-13	Deve ser guardado na base de dados do sistema os IDs dos cartões bem como o nível de acesso que cada cartão tem	Must Have	To improve	Out of the box	M	I

ID	Description	MoSCoW	As is/To be	Classification	Dev. Effort	I/NI
FR-14	Deve ser implementada uma base de dados com tabelas para os cartões registados, salas e acessos autorizados	Must Have	To improve	Out of the box	M	I
FR-15	Perfis de acesso devem incluir opções para acesso total, acesso departamental e restrições personalizadas	Should Have	To improve	Bespoke development	M	PI
FR-16	A funcionalidade de auditoria deve fornecer registros detalhados de todas as operações de abertura, incluindo data, hora e identificação do usuário	Could Have	To improve	Bespoke development	M	NI
FR-17	O sistema deve suportar cartões RFID padrão (por exemplo, MIFARE) para leitura pelo sensor RC522	Must Have	To improve	Out of the box	M	I
FR-18	O registro de novos cartões deve incluir associação de informações do usuário, como nome e departamento (opcional)	Should Have	To improve	Configuration	M	I
FR-19	O protótipo deve ser modular para facilitar futuras adições de novos sensores e dispositivos	Should Have	To improve	Out of the box	M	I
FR-20	O sistema deve incorporar um procedimento de backup automatizado para garantir a integridade dos dados.	Won't Have	To improve	Bespoke development	M	NI
FR-21	Quango é aberta uma sala são automaticamente ligadas as luzes e AC	Won't Have	To improve	Out of the box	M	NI
FR-22	Utilização de fechadura com *failsafe* mecânico	Must Have	To improve	Configuration	M	I

Tabela 2. - Requisitos funcionais

Foi também desenvolvida uma outra tabela com os requisitos não funcionais do sistema onde definimos alguns parâmetros esperados do sistema.

ID	Area	Description	Dev. Effort	I/PI/IN
NFR-1	**Compatibility**	O sistema deve comunicar sem problemas entre o Raspberry Pi 0 e o Raspberry Pi 4, mantendo a estabilidade e eficiência	M	I
NFR-2	**Compatibility**	A modularidade do código deve permitir fácil expansão para acomodar futuras atualizações e novas funcionalidades	L	I
NFR-3	**Availability**	O sistema deve manter um tempo de resposta abaixo de 1 segundo para garantir uma resposta rápida e eficiente	L	I
NFR-4	**Performance**	O consumo de energia do sistema deve ser otimizado para garantir uma operação eficiente, especialmente em caso de uso de bateria ou falta de energia	XL	NI
NFR-5	**Compatibility**	O sistema deve acomodar diferentes modelos de fechaduras solenoides sem grandes modificações no código ou na infraestrutura	XL	I

NFR-6	**Security**	As interfaces de hardware devem ser padronizadas para permitir futuras atualizações ou substituições sem interromper o funcionamento do sistema	L	I
NFR-7	**Purpose**	A interface de administração deve permitir fácil modificação de perfis de acesso e adição/remoção de usuários autorizados	M	PI

Tabela 3. - Requisitos não funcionais

4.1.Diagramas de Casos de Uso

O diagrama de caso de uso desenvolvido tem em conta todos os atores que interagem com o sistema, e sumariza as ações a que o mesmo deve responder.

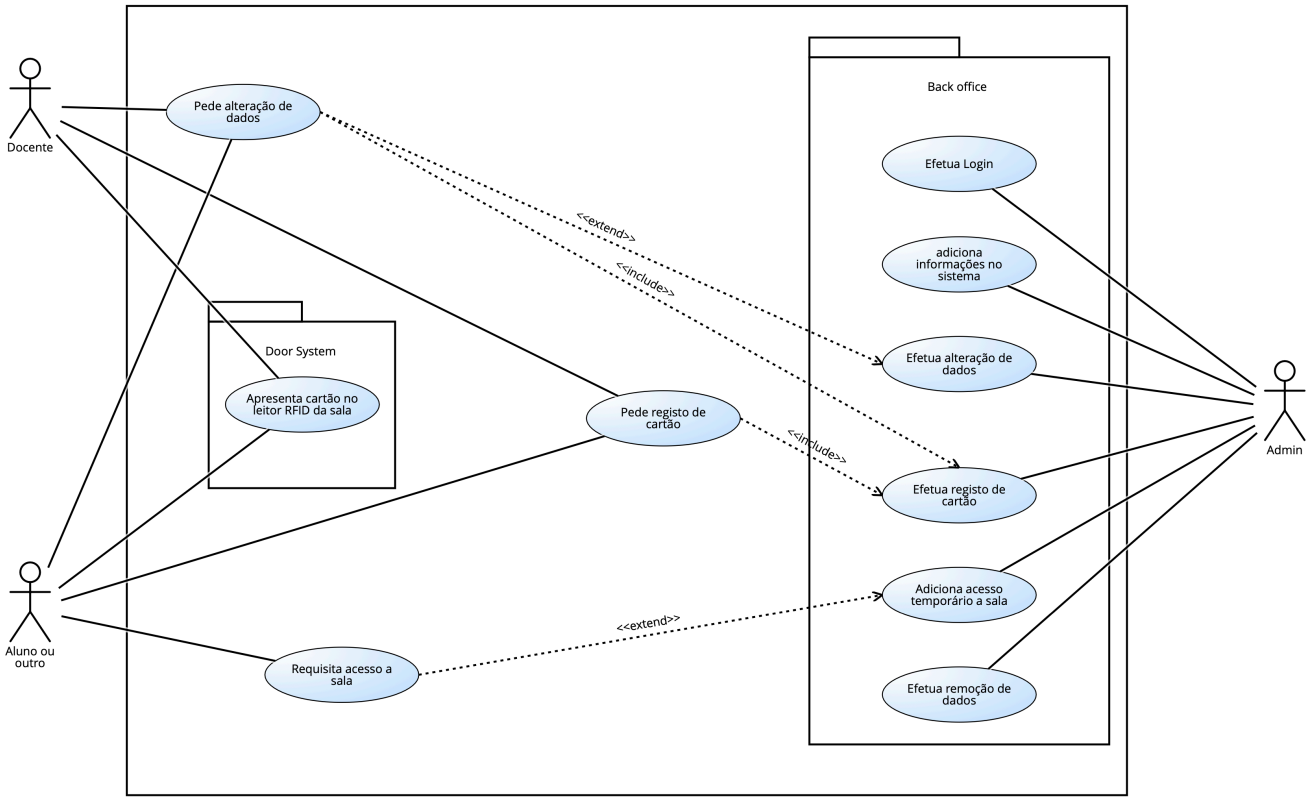


Figura 1. - Diagrama de Casos de Uso

Como evidenciado pelo diagrama grande parte da interação com a solução proposta vem por parte do administrador, isto porque desejamos que a interação por parte de docentes ou alunos seja muito intuitiva, onde uma vez registado o cartão no sistema o docente acede a sala de aula segundo a sua descrição e um aluno, mediante reserva prévia, tem acesso apenas á sala que lhe foi alocada.

Sendo que, como “admin”, entendemos como um descritivo para o login único que pretendemos implementar no back office, uma vez efetuado esse login o utilizador terá acesso ás funcionalidades do Back Office e poderá adicionar cartões ao sistema, alterar dados e acessos e remover cartões.

4.2.Diagramas de sequência e BPMN

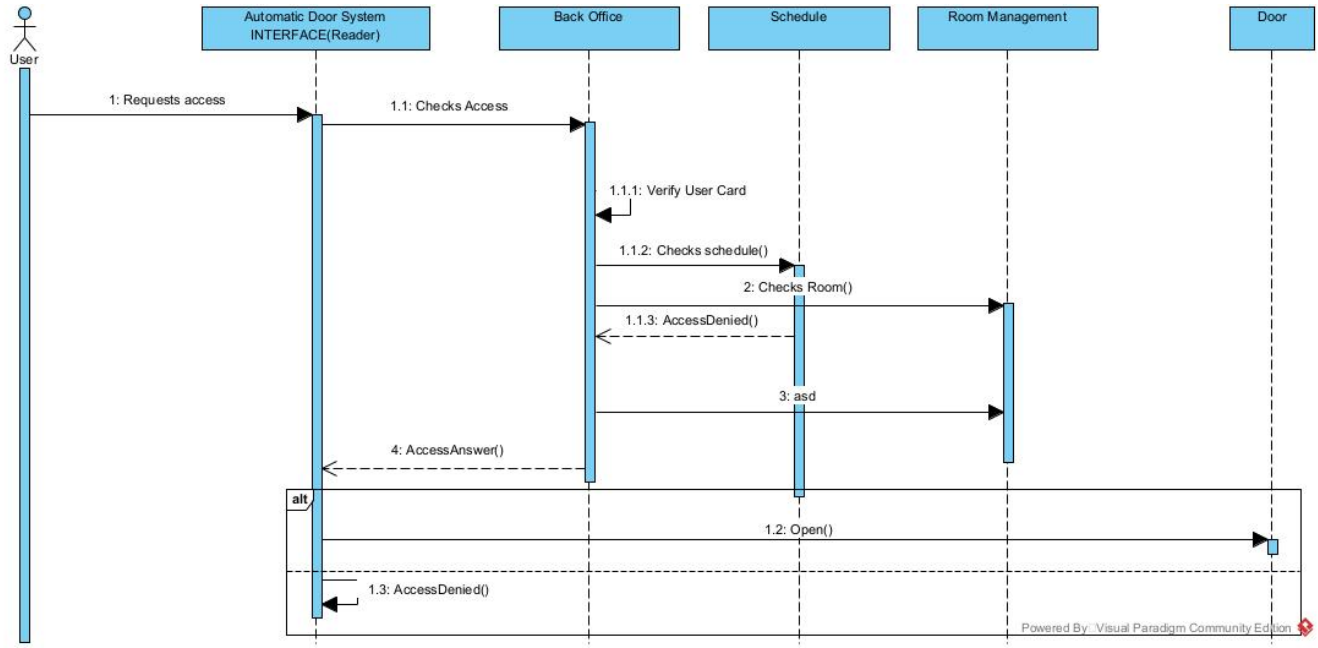


Figura 2. - Diagrama de Sequência - usar o sistema

Ao iniciar a solicitação de acesso, o User interage com o Sistema, de seguida o sistema verifica o horário do user, posteriormente o Sistema verifica também a atribuição de sala. Com base no que estiver no sistema, a porta será aberta ou não. Num caso positivo, a porta irá abrir/ destrancar.

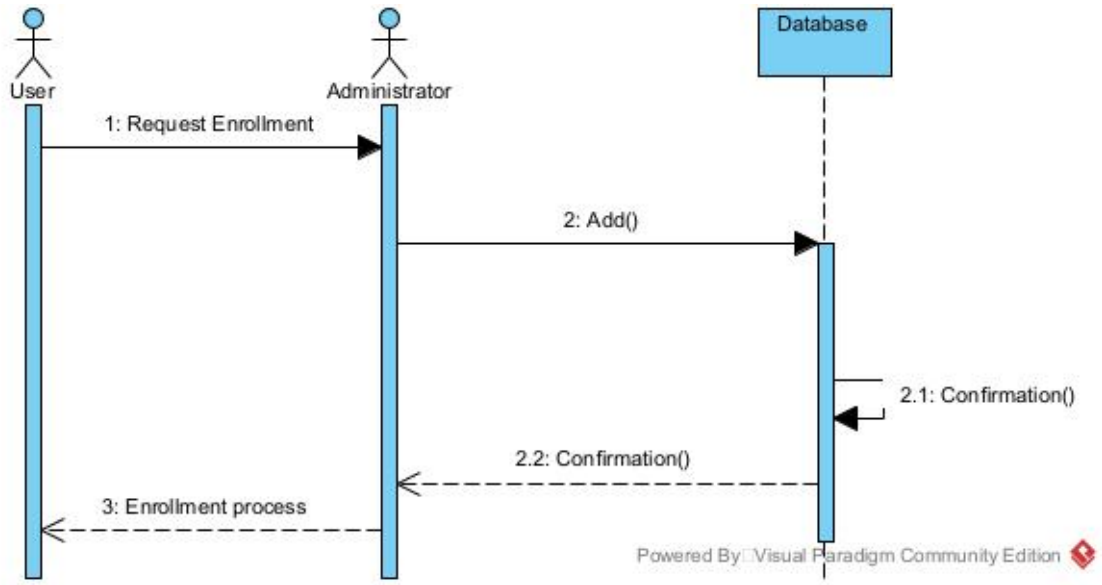


Figura 3. - Diagrama - Processo de inscrição

O User Inicia o processo de inscrição, diante do Administrador, fornecendo um cartão RFID, bem como informações adicionais, para estas serem adicionadas ao Banco de Dados.

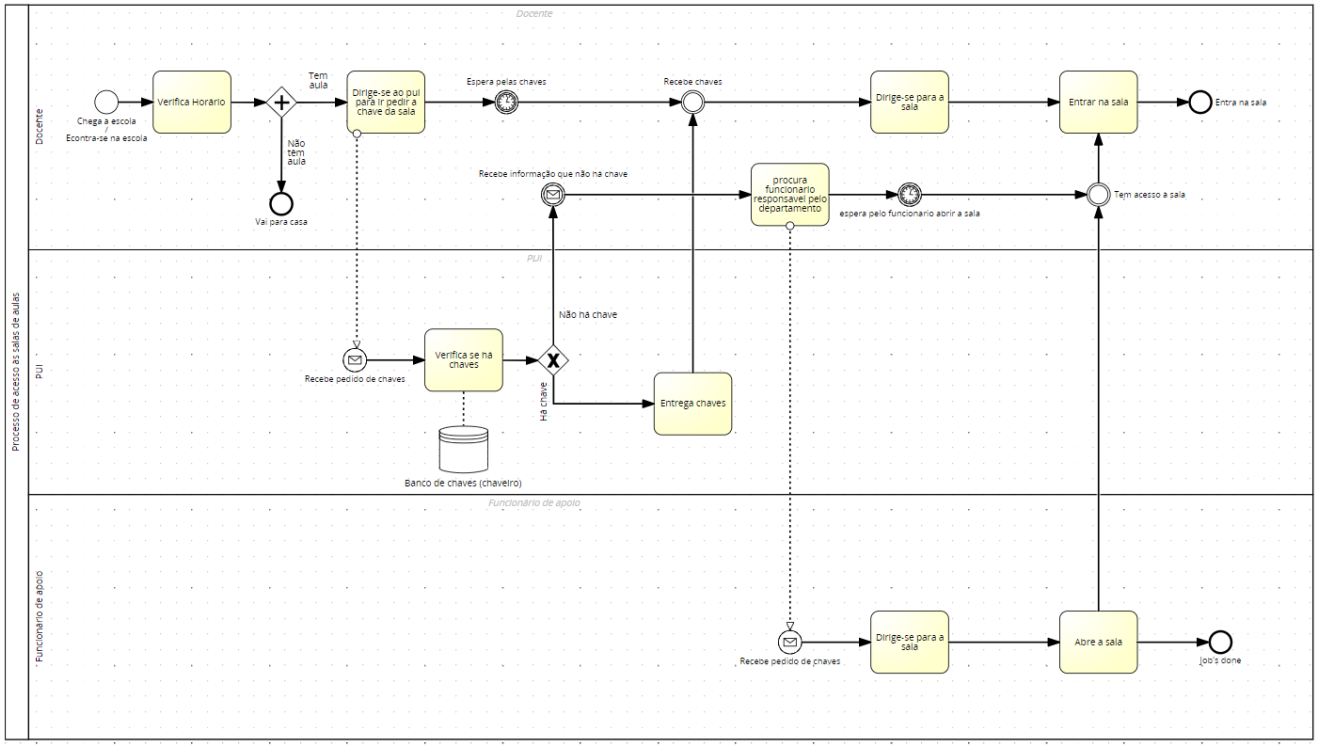


Figura 4. - Diagrama BPMN As-Is

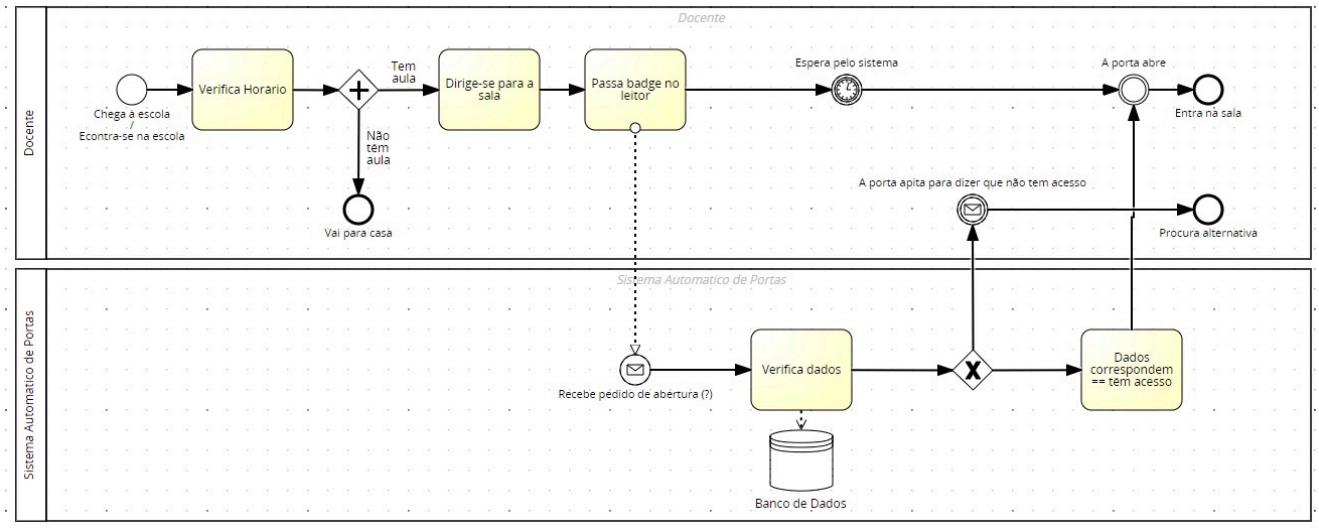


Figura 5. - Diagrama BPMN To-Be

Para demonstrar alguns dos casos de uso anteriormente identificados irá ser utilizado este diagrama de BPMN onde é apresentado todo um processo inerente ao casos de uso.

4.3.Modelos relevantes

A Solução proposta é sustentada por uma base de dados construída para guardar as informações dos usuários, como por exemplo o nome, id do cartão ou nível de acesso, que será um dos fatores que determina a abertura de uma porta. A tabela "classroom" guarda informações relevantes para categorizar uma sala de aula, de modo que as diferentes categorias vão interligar com os níveis de acesso. Na tabela "permissions" é delineada uma estrutura para o modo como o acesso de alunos ás salas de aulas, por meio de requisição, poderá ser feito. Sendo também uma possibilidade que esta tabela seja populada com informações relativas aos horários dos docentes. A tabela "access_level" será fixa e terá os seguintes dados:

id	Nome	Descrição
0	admin	Acesso total: Portas, Back-office
1	operator	Operador Back-office: Acesso total: portas Acesso parcial: Back-office
2	user-professor	user tipo 1: Acesso total: Portas Sem acesso: Back-office
3	user-student	user tipo 2: Acesso parcial: Portas Sem acesso: Back-office

Tabela 4. - dados "access_level"

Estes dados vão permitir a identificação do tipo de utilizador no sistema, uma vez que cada tipo de utilizador tem privilégios diferentes.

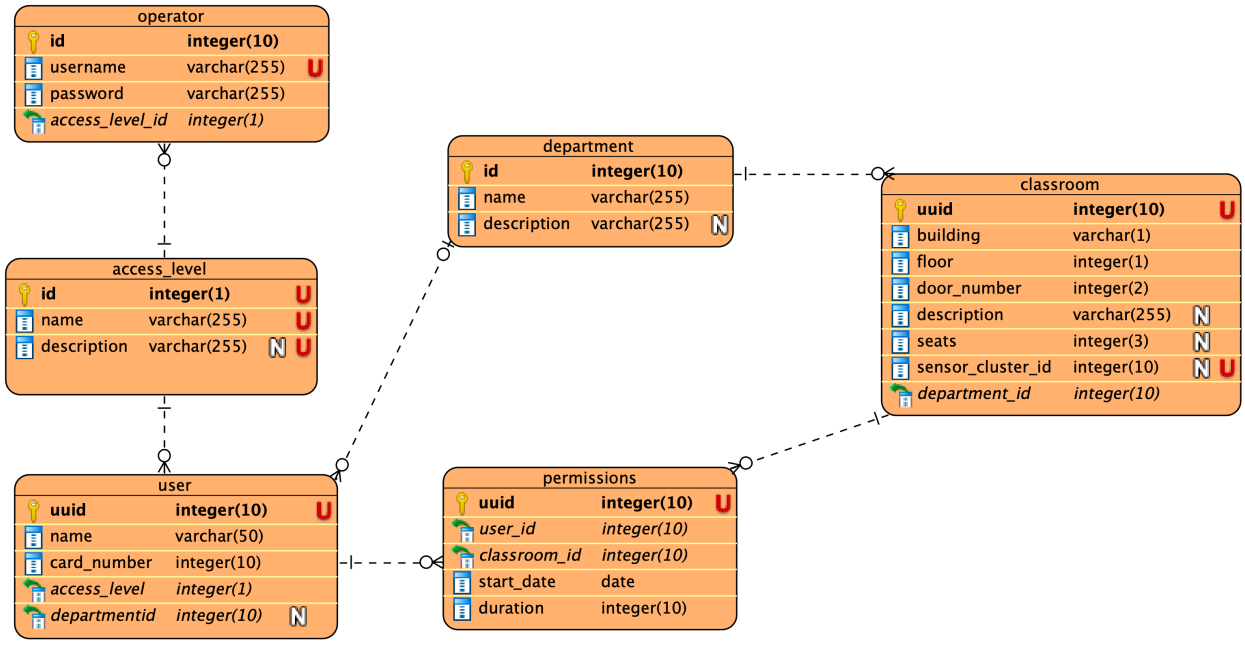


Figura 6. - Modelo da Base de Dados

4.4.Estrutura

Para idealizar a solução proposta é apresentada uma figura do aspeto base geral do sistema, porem é necessário expandir essa representação, por isso, foi então desenvolvido um diagrama da arquitetura do sistema.

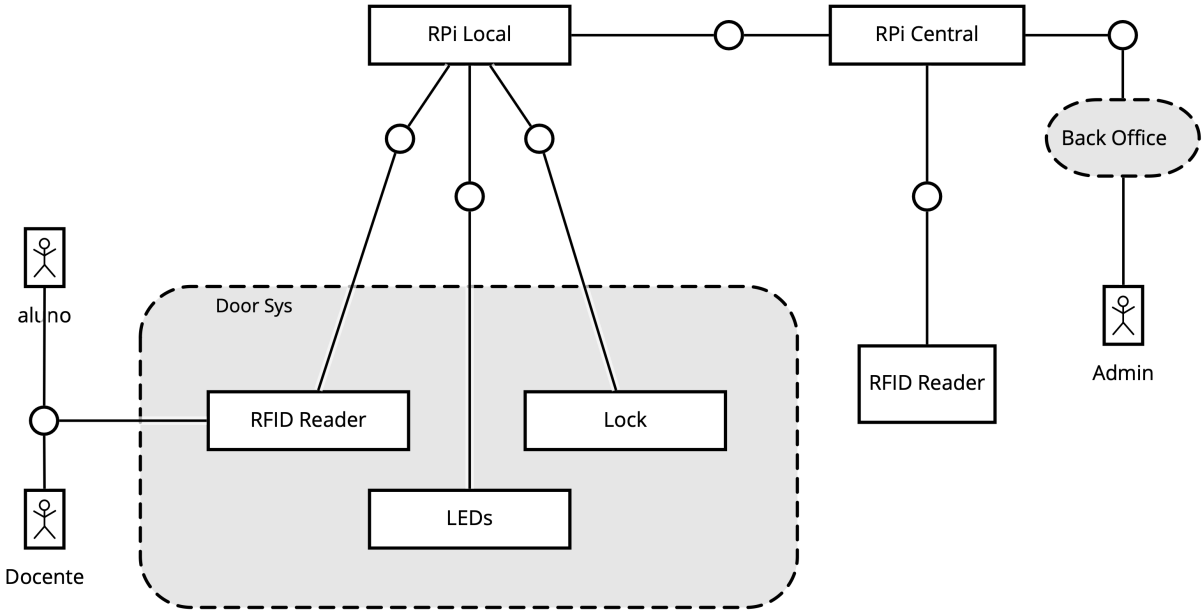


Figura 7. - Diagrama de Arquitetura do Sistema

No diagrama é apresentado uma versão reduzida comparativamente à solução que é desejada ser implementada, uma vez que a solução implementada é para ser replicada em todas as portas da faculdade, contudo neste diagrama é apresentado apenas uma porta (representado como "Door Sys"), mas seguem todas o mesmo sistema de implementação. Sendo que cada Raspberry Pi poderia suportar até 3 portas. É também demonstrado a necessidade de implementar um leitor RFID adicional que será utilizado como leitor para efeitos de registo, alteração ou remoção de cartões do sistema.

4.5.Mock ups

Foi desenvolvido um mock up que serviu de guia para a implementação do Back-office. Inicialmente será apresentado uma pagina de login, onde o operador (previamente introduzido na BD) introduz o nome de utilizador e palavra-chave.

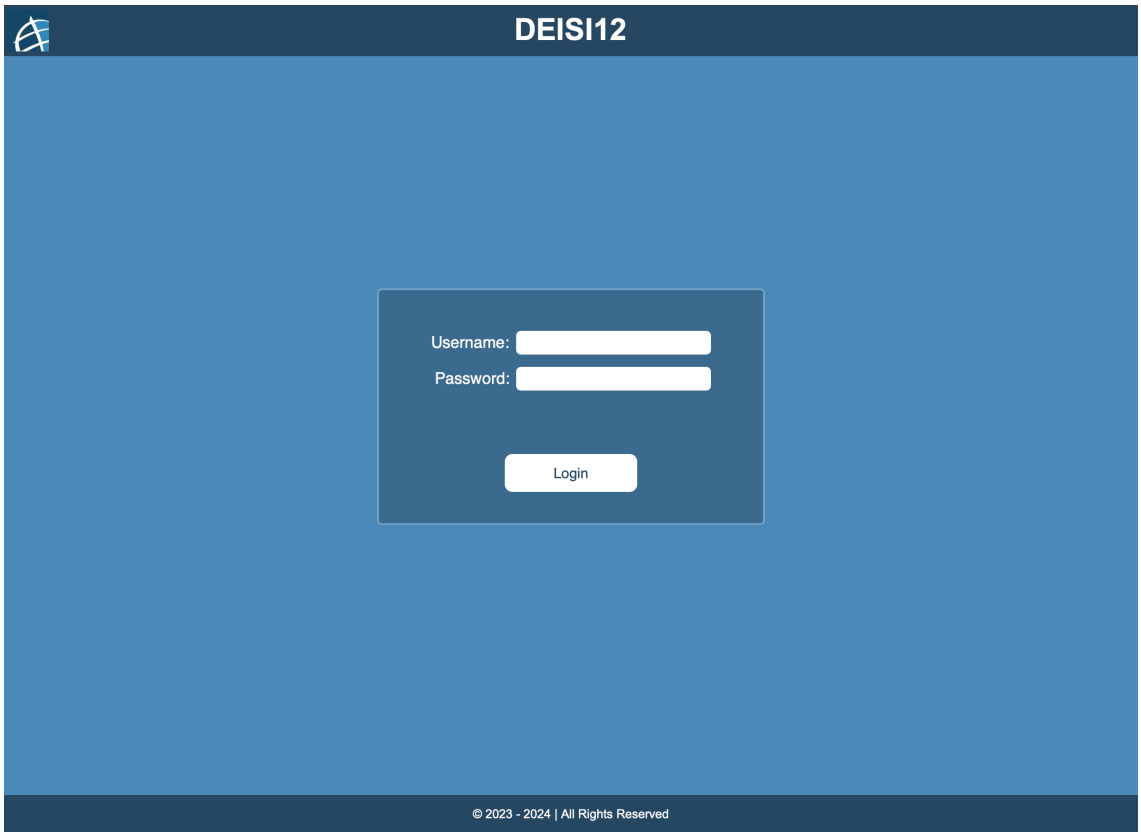


Figura 8. - Login

Seguidamente, é apresentada a página principal onde o operador pode ver os acessos ás salas e realizar operações relacionadas com os utilizadores.

Figura 9. - Página principal

Qualquer uma das interações vai levar o operador a passar um cartão num leitor RFID, após isso o ID do cartão é preenchido no formulário que será depois submetido á base de dados.[2]

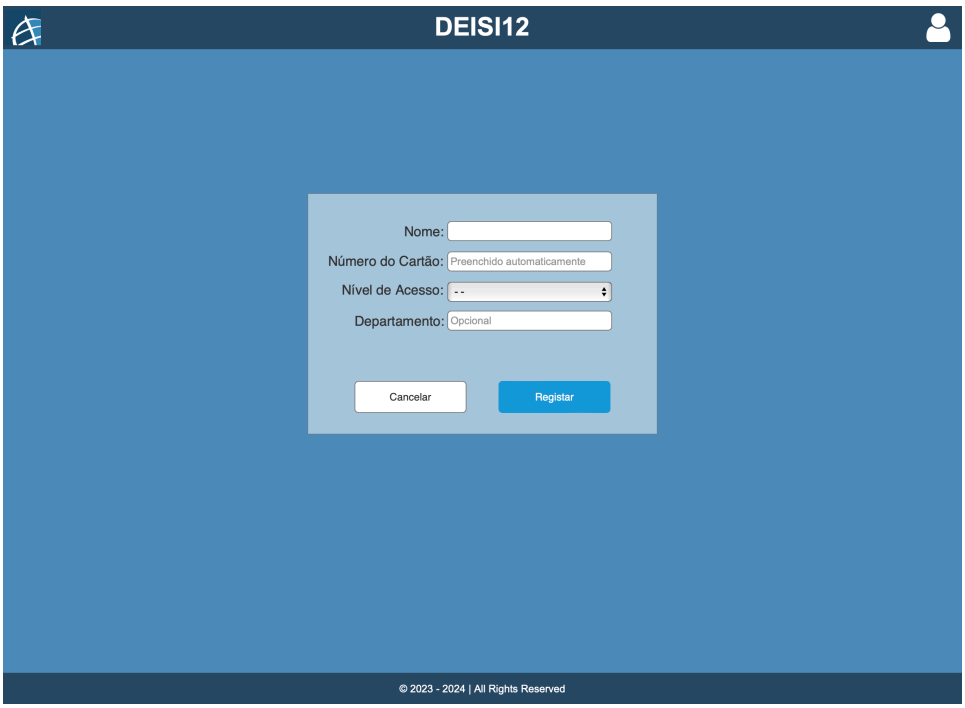


Figura 11. - Formulário de registo de utilizador

[2] Link para acesso ao Mock Up: https://eeovzf.axshare.com/?code=b180442eb690be98219e41f4588f6f78

4.6.Modelo 3D

Para a realização da maquete final foi desenvolvido o seguinte modelo 3D, que foi depois cortado a laser em madeira prensada, nas salas do LabTech da Universidade Lusófona. Apesar do modelo real não seu totalmente idêntico ao modelo apresentado na Figura 12. O mesmo difere apenas nos quadrados em cada canto que iriam ser inicialmente usados para segurar o modelo a base.

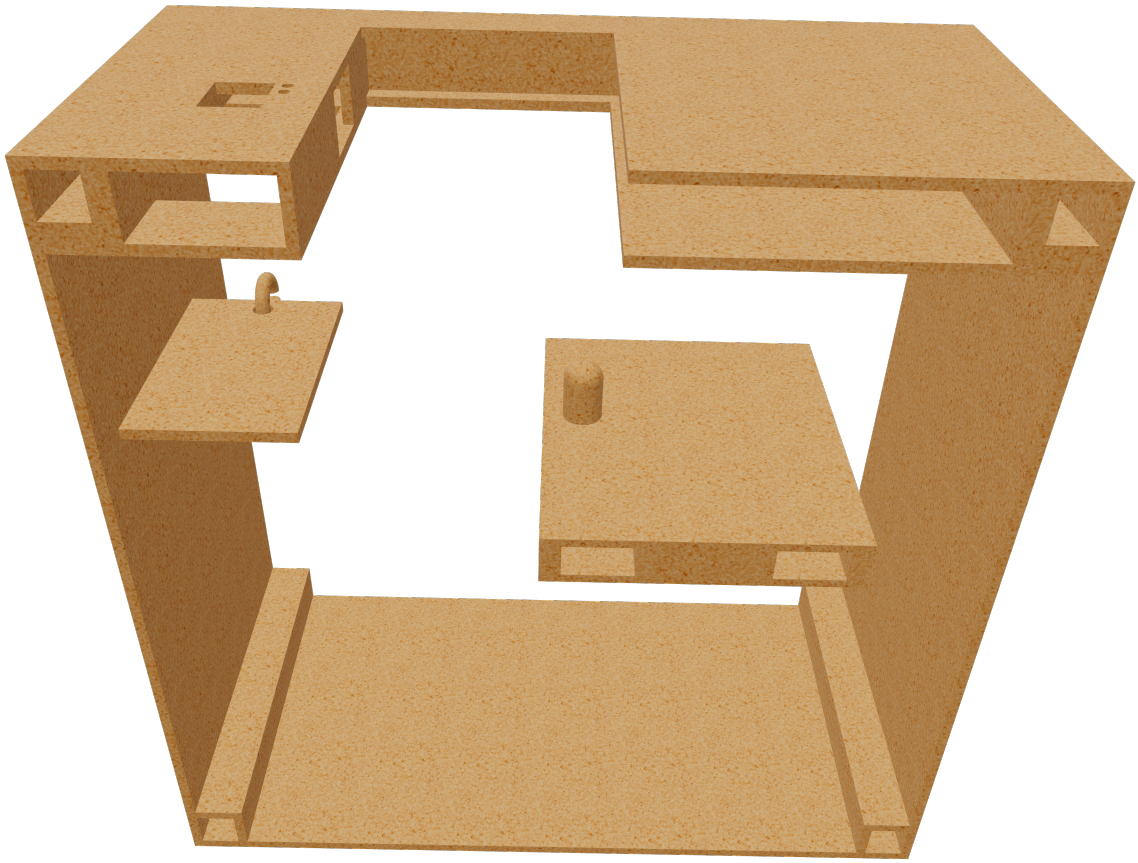


Figura 12. - Modelo 3D[3]

[3] Link para visualização modelo 3D: https://collaborate.shapr3d.com/v/WuswEJWoPkSEdd5gwOLM-

5. Solução Proposta

5.1.Introdução

O sistema proposto foi dividido em duas fases de desenvolvimento, primeiramente o hardware necessário para desenvolver um protótipo e em segundo o desenvolvimento do software que irá fazer uso dos componentes. Um sistema que permite aos utilizadores realizar o acesso as salas de aula através de um cartão com funcionalidades RFID. Suportado por um servidor com uma base de dados que contem quais os utilizadores que têm acesso e será também host de um website que permita fazer a gestão de utilizadores e acessos.

Links:

- Youtube (Video demonstrativo): https://youtu.be/-gJ8qVmDD2w?si=Y8g3Do40a2vIqNlZ
- Repositório GitHub: https://github.com/DEISI-ULHT-TFC-2023-24/TFC-DEISI12-Sistemas-de-abertura-de-portas-das-salas-de-aulas.git

5.2.Arquitetura

No hardware, para ser efetuada a leitura de cartões é usado um sensor RFID, mais especificamente o RC522. Ainda não está claro a fechadura a ser apresentada na solução de mercado, têm de ser consideradas várias fechaduras com *Failsafe* mecânico. Tendo em consideração uma das linguagens de programação a ser usadas no projeto, Python, foi escolhido usar computadores de placa única desenvolvidos pela Raspberry Pi Foundation, nomeadamente o Raspberry Pi 0 e Raspberry Pi 4, que vão trabalhar em conjunto para concretizar as funcionalidades do sistema. Estes comunicam via Wi-Fi através de um Access Point, passando as informações necessárias para o funcionamento via comunicação UDP.

A figura 1. apresenta o diagrama base que a solução desenvolvida deverá seguir, sendo que o sistema porta está incluido o sensor RFID, fechadura e LEDs como mencionado anteriormente.

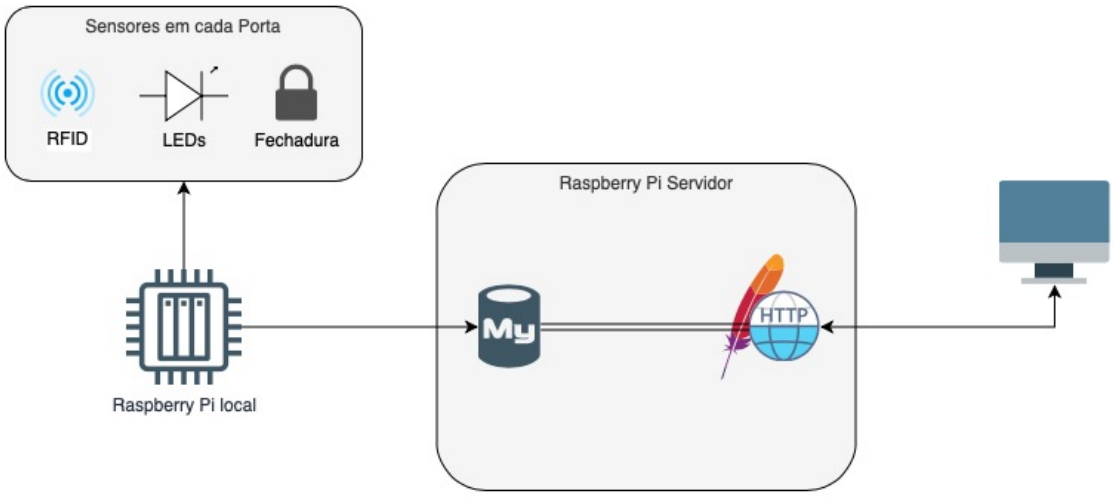


Figura 13. - Diagrama do sistema

5.3.Tecnologias e ferramentas utilizadas

O sistema fisico implementado é suportado por código desenvolvido principalmente em python, onde foram desenvolvidas as funcionalidades de leitura pelo sensor RFID, a abertura do fecho e confirmação de acesso com LEDs, a gestão de acessos é feita usando uma unidade de processamento central (Raspberry Pi, como servidor) ligada a uma base de dados desenvolvida no âmbito deste projeto. Esta base de dados é preenchida com a implementação um Back-office com a capacidade de registar cartões no sistema e a esses serem atribuídos um nível de acesso.

Juntando estas duas componentes o funcionamento do protótipo segue o seguinte funcionamento: é apresentado um cartão com chip RFID a um sensor RFID , este comunica com o Raspberry Pi alocado aquele conjunto de portas que comunica com outro Raspberry Pi (servidor) onde é feita a verificação de acesso, caso positivo a porta é aberta, caso negativo a porta não será aberta.

Os componentes a ser usados que achamos serem os ideais para este projeto são:

Fechadura solenóide: A ideal para ser implementada será uma fechadura que permita a sua abertura não só por meios eletrónicos, mas também por meios convencionais (chave), em caso de falha energética. Porem é também essencial que tenha um fecho mecânico na parte que fica virada para dentro de uma sala, por motivos de segurança. Para a maquete que será apresentada, foi escolhida a fechadura presente na Figura 14., que apresenta o controlo eletrónico e o fecho mecânico na parte interior.

Figura 14. - Fechadura solenoide

Raspberry Pi Zero (Raspberry Pi-0): Um computador de placa única pequeno acessível e de baixo consumo de energia, com o sistema operativo Debian, desenvolvido pela Raspberry Pi Foundation. Commumente utilizado em projetos de bricolage (faça você mesmo), aplicações de IoT (Internet das Coisas) e sistemas embarcados.

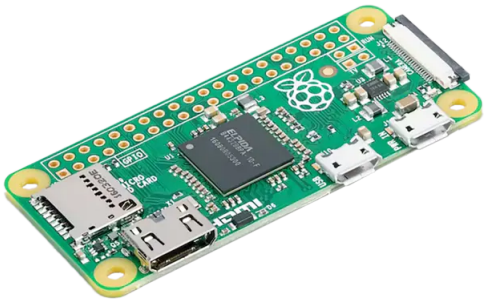


Figura 15. - Raspberry Pi 0

RFID Card: Um cartão inteligente com uma etiqueta RFID incorporada que utiliza sinais de radiofrequência para comunicação. Commumente usado para controlo de acesso, identificação e rastreamento.

RFID Reader (RC522): Um dispositivo que lê informações de etiquetas ou cartões RFID utilizando sinais de radiofrequência. Commumente usado em sistemas de controlo de acesso, gestão de inventário e rastreamento.

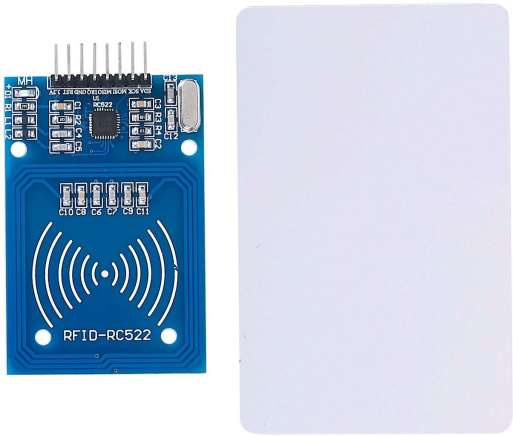


Figura 16. - Leitor RFID RC522 e Cartão

Python: Linguagem de programação de alto nível utilizada em várias áreas da ciência da computação, como análise de dados, programação web ou automação de tarefas.

SQLite: Uma biblioteca de linguagem C que implementa um mecanismo de base de dados SQL pequeno, rápido, autónomo, altamente confiável e cheio de funcionalidades.

Para a base de dados será utilizada a ferramenta MariaDB que faz uso da linguagem SQL para a criação de modelos relacionais. O Back-office foi implementado utilizando Flask (Python), HTML, CSS, JavaScript.

5.4.Implementação

A próxima fase de desenvolvimento é um protótipo inicial com recurso a sensores LED, para testar o código desenvolvido e assim dar inicio á programação das funcionalidades base. No protótipo atualmente a ser desenvolvido se a identificação do RFID que for apresentada no leitor estiver presente na base de dados, então é garantido acesso com acender do LED verde, caso contrario irá ativar um LED vermelho. Seguidamente a esta funcionalidade base e critica

do sistema estar implementada serão adicionadas as restantes funcionalidades, bem como a implementação dos sensores numa maquete funcional.

O protótipo consiste num controlador, que incorpora um Raspberry Pi conectado a uma placa de ensaio através de um extensor GPIO, onde estão ligados um leitor RFID, dois LEDs (Verde e Vermelho) e um *relay* outrora conectado a uma fechadura.
Ao correr o programa, quando apresentado um cartão diante do leitor de RFID, é enviada uma mensagem para o Raspberry Servidor, que mediante a resposta devolvida, é então enviado um sinal eléctrico para a fechadura abrir, caso seja garantido acesso.

No Raspberry Pi Servidor está configurada as instancia de base de dados, usando a ferramenta MariaDB, que usa linguagem SQL. No Anexo 2. é apresentado o código SQL que origina a base de dados descrita.

Na Figura 17. é apresentado um esquema de ligações para o correto funcionamento do sistema uma vez instalado. Por ordem descendente os sensores são: fechadura e relay, os cabos sem ligação são os de fonte de alimentação a fechadura, Sensor RFID, LED vermelho e LED verde.

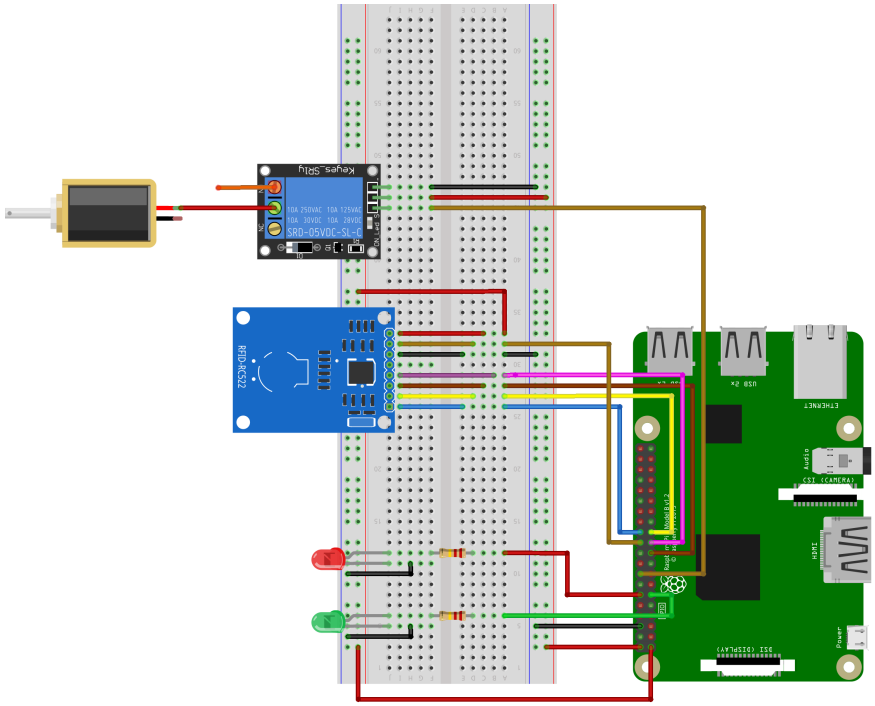


Figura 17. - Esquema de ligação Raspberry Local

5.5.Abrangência

Este trabalho apresenta vários conteúdos relacionados com o que aprendemos durante o curso. Python foi uma linguagem lecionada em Data Science. Através de arquitetura de computadores e arquitetura avançada de computadores, conseguimos entender os desafios e respetivas soluções enquanto trabalhamos com os Raspberry Pi. Analise e Concepção de Sistemas e Sistemas de Suporte e Decisão permitiu-nos ter os fundamentos teóricos e práticos sobre os diagramas desenvolvidos ao logo do projeto. A cadeira de base de dados está presente na criação da base de dados do sistema. A criação do mock up foi baseada nos tópicos aprendidos na cadeira de interação humano-máquina e a comunicação utilizada entre Raspberry Pi lecionada em Redes de Computadores.

6. Plano de testes e avaliação

Tendo completado a programação dos sensores em uso e a configuração do Raspberry Pi que vai servir como servidor, passamos à realização dos testes seguintes. O primeiro teste foi o de registo de novos utilizadores/cartões e acesso imediato. O objetivo era verificar se novos utilizadores podem ser registados e se têm acesso imediato após o registo. Os testes foram concluídos com sucesso: novos utilizadores foram registados e conseguiram acesso imediato às salas, com a base de dados sendo atualizada instantaneamente.

Seguidamente, implementámos e testámos vários tipos de acesso. O objetivo era testar diferentes tipos de acesso e os resultados mostraram que todos funcionaram conforme esperado. O sistema reconheceu cartões válidos e tags válidas, com vários níveis de acesso.

Testámos também o acesso indevido de um aluno/user que não requisitou sala. O objetivo era garantir que alunos/users sem autorização não conseguissem acesso. O sistema negou corretamente o acesso a alunos não autorizados, com LEDs vermelhos acendendo para indicar o acesso negado.

Outro teste foi o de comunicação entre dois Raspberry Pi. O objetivo era testar a comunicação entre um Raspberry Pi local (controlador) e um Raspberry Pi servidor. A comunicação foi bem-sucedida utilizando um router em modo AP, com mensagens enviadas e recebidas corretamente, demonstrando que o sistema pode operar de forma integrada e sincronizada.

Implementámos também a base de dados e o back-office. O objetivo era verificar a adição, alteração e remoção de utilizadores através do back-office. O back-office foi acedido via browser, e as operações de gestão de utilizadores (adição, alteração e remoção) foram refletidas corretamente na base de dados e no sistema de portas.

Os testes realizados demonstram que a solução cumpre os objetivos a que se propôs, contribuindo eficazmente para a resolução de um problema real. Os resultados comprovam a aplicabilidade, pertinência e relevância do sistema desenvolvido. Abaixo estão os resultados organizados para demonstrar a implementação e operacionalidade dos requisitos trabalhados ao longo do desenvolvimento do projeto.

A funcionalidade do sistema de abertura de portas foi testada inicialmente através de "*hard coding*", mostrando que o sistema responde corretamente à apresentação de cartões RFID, abrindo a fechadura e acionando LEDs conforme esperado. A comunicação e integração entre dois Raspberry Pi utilizando um router em modo AP foi validada, com mensagens codificadas enviadas e recebidas, confirmando a funcionalidade de comunicação do sistema.

A gestão de acessos foi testada através do registo de novos utilizadores e gestão de acessos pelo Back-office, mostrando que o sistema é robusto e eficiente. As alterações feitas no Back-office foram refletidas em todo o sistema. Cenários reais de utilização foram simulados, incluindo tentativas de acesso por alunos não autorizados e a resposta do sistema a falhas electrónicas. A participação de docentes na fase de testes proporcionou *feedback* relevante e valioso.

Durante os testes, foram identificadas e implementadas várias melhorias, como a otimização do processo de registo de utilizadores e melhorias na interface do Back-office para facilitar a adição e gestão de utilizadores.

7. Método e Planeamento

Tendo todas as fases concluídas podemos aferir que os objetivos foram cumpridos e solução desenvolvida cumpre as funcionalidades a que se propôs.

12	**Fase 4 - Entrega Final**	2024-04-19	2024-06-28
35	Implementação de funcionalidades do sistema	2024-04-19	2024-05-09
31	Desenvolvimento de maquete final	2024-05-01	2024-05-20
34	Implementação de Back Office	2024-05-09	2024-06-02
36	Testes - Back Office	2024-05-28	2024-06-21
37	Testes - Maquete Final	2024-05-28	2024-06-21
15	Atualização do Relatório	2024-06-15	2024-06-24
14	Entrega Final	2024-06-28	2024-06-28

tabela 5. - Calendário

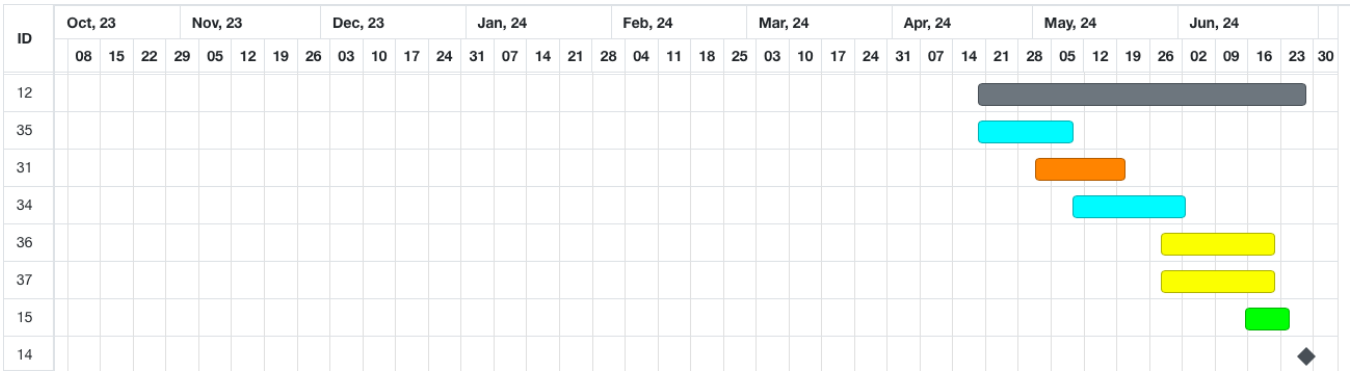


Figura 18. - Calendário Gantt

A tabela e figura apresentam as tarefas associadas ao segundo semestre, considerando o Anexo 4.

Nesta fase de desenvolvimento, focou-se em concluir o código para a interação dos Raspberry Pi com o back-office, implementando todas as funcionalidades necessárias para o término do sistema. Foram realizados testes de cada componente, seguidos por testes reais com uma maquete para verificar a comunicação entre os componentes. Além disso, testes de usabilidade foram conduzidos com a participação de docentes e alunos para garantir que o sistema atendesse às suas necessidades.

Umas das alterações a ser feita para a utilização da solução desenvolvida no mercado é a de procuração das fechadura que se adeque as necessidades da Universidade.

O calendário Gantt inicial foi, em grande parte, seguido, embora tenham ocorrido alguns desvios. A fase de desenvolvimento foi concluída conforme planejado, enquanto a fase de testes necessitou de mais tempo devido a problemas imprevistos na comunicação entre os componentes. O esforço desenvolvido ao longo do ano foi significativo, com variações na

intensidade do trabalho conforme as diferentes fases do projeto. Durante a fase de testes, o esforço aumentou substancialmente para resolver problemas técnicos e realizar ajustes.

O projeto sofreu alterações devido à necessidade de implementar funcionalidades adicionais não previstas inicialmente, como a integração do back-office sendo implementado na rede local com a ajuda do Flask. O esforço inicial estimado foi subestimado em algumas áreas, especialmente na fase de testes e ajustes. Desafios técnicos na comunicação entre componentes e na integração de novos métodos de acesso foram superados com trabalho adicional e a colaboração da equipa.

8. Resultados

Os resultados obtidos vão ao encontro da solução proposta inicialmente, o sistema final apresentado tem a capacidade de abrir salas a docentes, funcionários e alunos mediante requisição previa. O Back-office implementa as funcionalidades de login no website, permite registar cartões, alterar dados e a remoção de cartões. E é também no Back-office que é Implementada a funcionalidade de requisitar uma sala para alunos.

O código desenvolvido está preparado para ter vários sensores de varias portas, mas para a demonstração apresentada é apenas utilizado um conjunto de sensores representando uma sala, mas sem ser necessário esforço adicional o sistema é escalável. Os nossos resultados demonstram que o sistema consegue diferenciar entre um cartão de funcionário/Docente e aluno e que consegue também permitir acesso a um aluno que tenha requisitado a sala.

O Back-office apenas se desvia na aparência estética face aos Mock Ups apresentados no ponto 4.5, cumprindo todas as funcionalidades propostas, sendo a sua aparência altamente costumizável utilizando CSS.

9. Conclusão e Trabalhos Futuro

Neste trabalho conseguimos aprender que uma conceptualização da solução proposta é vital para o seu desenvolvimento e implementação. Todos os modelos desenvolvidos tornaram possível ter uma visão clara dos passos a tomar e do esforço necessário para cada tarefa. A construção de uma maquete funcional tornou possível fazer uma demonstração e realizar testes ao sistema. O trabalho tem via de continuação futura, sendo possível desenvolver ainda mais o código com base em testes ao sistema usando múltiplos sensores e Raspberry Pi. O Back-office pode receber ainda mais funcionalidades e numa eventual expansão ser disponibilizado online aos estudantes para reservar salas diretamente na plataforma. Outra possível continuação deste trabalho é a implementação de pequenos ecrãs LCD ou E-Ink com informação relativa ao horário de cada sala.

Bibliografia

[DEISI23] DEISI, Regulamento de Trabalho Final de Curso, Set. 2023.

[TaWe20] Tanenbaum,A. e Wetherall,D., *Computer Networks*, 6ª Edição, Prentice Hall, 2020.

[ULHT23] Universidade Lusófona de Humanidades e Tecnologia, www.ulusofona.pt, acedido em Out. 2023.

[DRAWIO] https://www.drawio.com/blog/sysml-vs-uml, acedido em Jan. 2024

[G99] https://www.guru99.com/uml-cheatsheet-reference-guide.html, acedido em Jan. 2024.

[SigAcad] https://academic.signavio.com/, acedido em Jan. 2024.

[UMLD] https://www.uml-diagrams.org/use-case-diagrams-examples.html, acedido em Jan. 2024.

[MySQL] https://dev.mysql.com/doc/refman/8.0/en/, acedido em Fev. 2024

[MySQL] https://dev.mysql.com/doc/ , acedido em Mar. 2024

[RPi] https://raspberrytips.com/install-mariadb-raspberry-pi/ , acedido em Mar. 2024.

[RPi] https://raspberrytips.com/web-server-setup-on-raspberry-pi/ , acedido em Mar. 2024.

[GFG] https://www.geeksforgeeks.org/differences-between-tcp-and-udp/ , acedido em Mar. 2024.

[Python] https://wiki.python.org/moin/UdpCommunication , acedido em Mar. 2024

[MDB] https://mariadb.com/resources/blog/how-to-connect-python-programs-to-mariadb/ , acedido em Mai. 2024

[MDB] https://www.raghwendra.com/blog/how-to-connect-html-to-database-with-mysql-using-php-example/ #Step_4_Create_a_PHP_page_to_save_data_from_HTML_form_to_your_MySQL_database , acedido em Mai. 2024

[MDB] https://mariadb.com/resources/blog/developer-quickstart-php-mysqli-and-mariadb/ , acedido em Mai. 2024

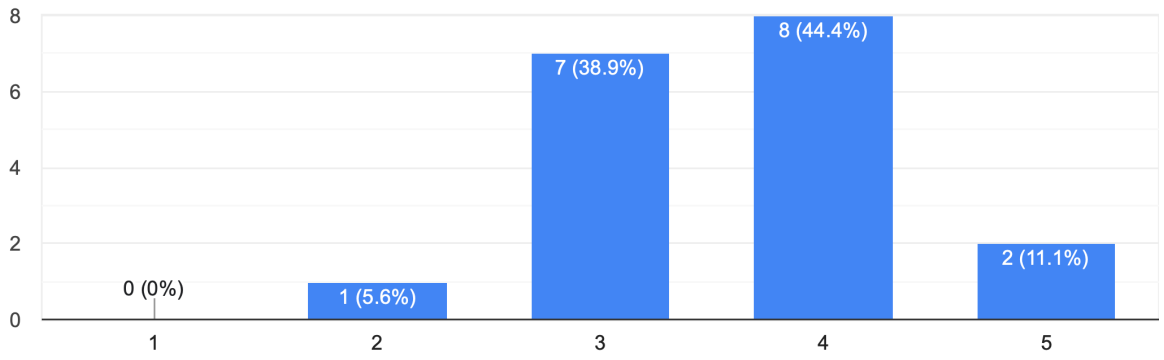
[RPi] https://www.raspberrypi.com/documentation/computers/os.html#python-on-raspberry-pi , acedido em Jun. 2024

[SPR3D] https://www.shapr3d.com/ , acedido em Jun. 2024

Anexo 1 - Questionário

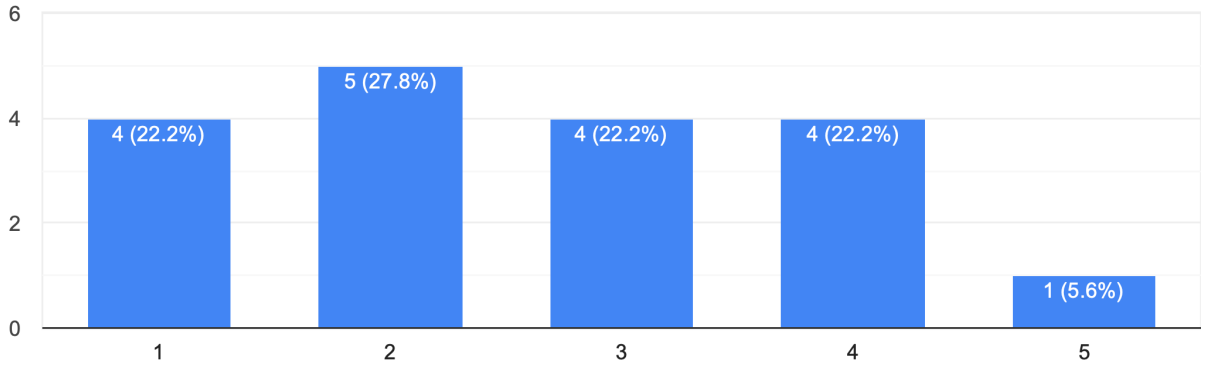
Na requisição e entrega de chaves como avalia o processo relativamente a sua rapidez?

18 responses



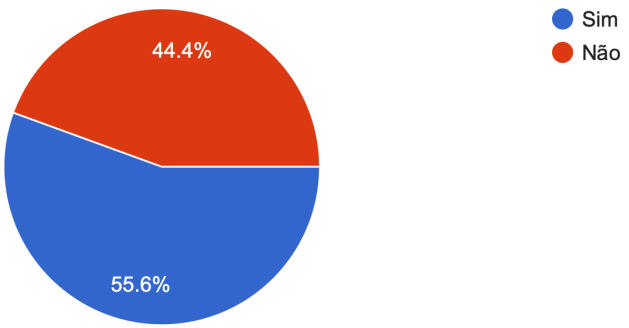
Com que frequência teve de esperar para obter chaves do PUI

18 responses



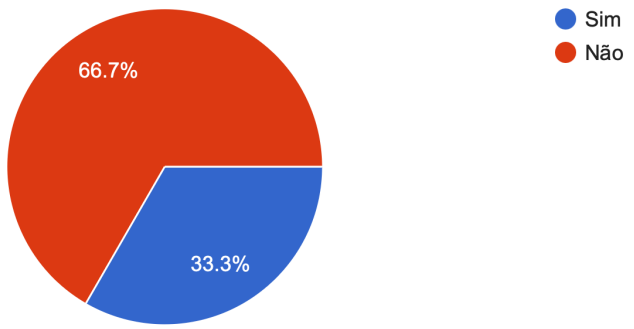
Alguma vez foi necessário esperar que a chave que necessita fosse entregue no PUI?

18 responses



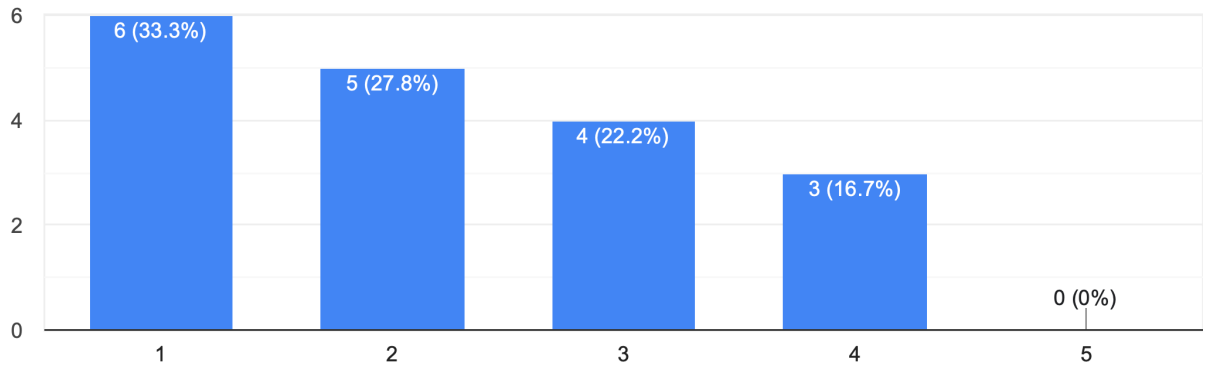
Alguma vez teve dificuldade em usar as chaves na abertura das portas?

18 responses



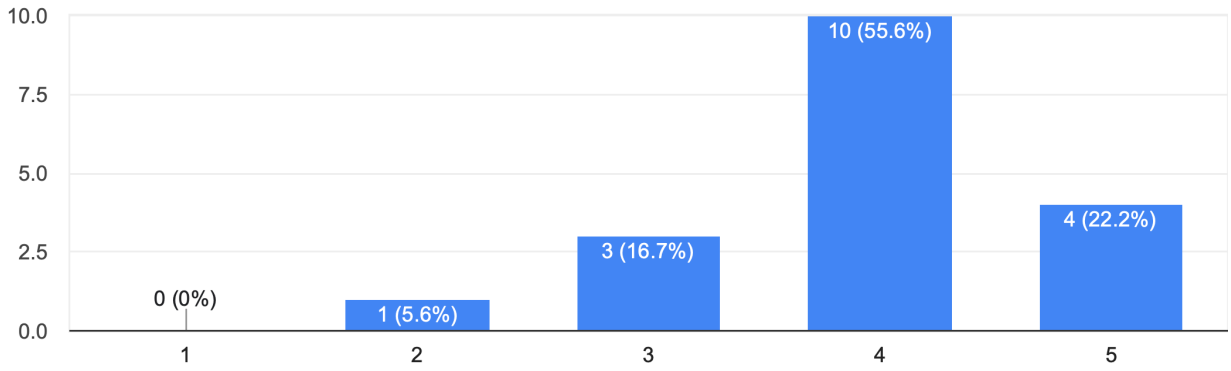
Quão inconveniente é ter de se deslocar ao PUI sempre que necessita de trocar de sala?

18 responses



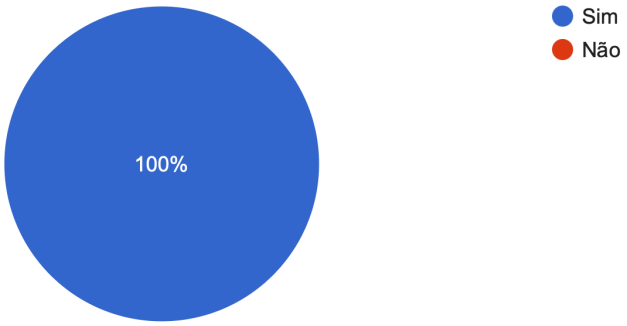
Na generalidade, está satisfeito/a com os serviços prestados pelo PUI?

18 responses



Se existisse sistema que permitisse a não deslocação ao PUI para Requisição e entrega de chaves, iria querer adotar esse sistema?

18 responses



Caso o sistema seja aplicado em toda a faculdade, quais as funcionalidades que gostaria de ver implementadas?
Abrir porta a partir da app docentes Lusofona, nomeadamente a partir do horario, sem necessidade de usar cartao.
Entradas/saídas em/de laboratório
Acesso às salas. sistema de requisição de salas. sistema de lançamento de notas.
Sumários lançados automaticamente e os alunos através de um QR code lançassem as presenças
Resolver os problemas apontados neste questionário
Gostaria que o acesso fosse através de PIN ou cartão de modo que pudesse ser realizado o acesso independente do meio com que o docente entrou na sala com facilidade no caso de troca de docente. É comum esperar que um dado docente termine sua aula para que outro entre na sala e comece a sua. Se houver a necessidade do primeiro ocupante registar a saída da sala para então o segundo ocupante dar entrada, pode ocasionar transtornos e perda de tempo uma vez que o mesmo já ocorreu diversas vezes com chaves.
Abertura de salas, requisição de comandos de datashow e AC
Abertura com cartão; Requisição de salas via web/app; Ver ocupação salas via web/app

nada a acrescentar

Anexo 2 - Código SQL

```
--
-- Table structure for table `access_level`
--

CREATE TABLE `access_level` (
  `id` int(10) NOT NULL,
  `name` varchar(255) NOT NULL,
  `description` varchar(255) DEFAULT NULL
)

--
-- Dumping data for table `access_level`
--

INSERT INTO `access_level` (`id`, `name`, `description`) VALUES
(0, 'admin', 'full access: doors, back-office'),
(1, 'operator', 'back-office operator\r\nfull access: doors\r\npartial access: back-
office'),
(2, 'user-professor', 'user type 1\r\nfull access: doors\r\nno access: back-office'),
(3, 'user-student', 'user type 2\r\npartial access: doors\r\nno access: back-office');

-- --------------------------------------------------------

--
-- Table structure for table `classroom`
--

CREATE TABLE `classroom` (
  `id` int(10) NOT NULL,
  `building` varchar(1) NOT NULL,
  `floor` int(1) NOT NULL,
  `door_number` int(3) NOT NULL,
  `description` varchar(255) DEFAULT NULL,
  `department_id` int(11) DEFAULT NULL,
  `seats` int(3) DEFAULT NULL,
  `sensor_cluster_id` int(10) NOT NULL
)

-- --------------------------------------------------------

--
-- Table structure for table `department`
--

CREATE TABLE `department` (
  `id` int(10) NOT NULL,
  `name` varchar(255) NOT NULL,
  `description` varchar(255) DEFAULT NULL
)
```

```
-- -------------------------------------------------------

--
-- Table structure for table `operator`
--

CREATE TABLE `operator` (
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `access_level_id` int(10) NOT NULL
)

--
-- Dumping data for table `operator`
--

INSERT INTO `operator` (`username`, `password`, `access_level_id`) VALUES
('joao.joyce', 'Admin0_BO', 0),
('test-operator', 'Operator', 1);

-- -------------------------------------------------------

--
-- Table structure for table `permission`
--

CREATE TABLE `permission` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `classroom_id` int(11) NOT NULL,
  `start_date_time` datetime(6) NOT NULL,
  `duration` time(3) NOT NULL
)

-- -------------------------------------------------------

--
-- Table structure for table `user`
--

CREATE TABLE `user` (
  `id` int(10) NOT NULL,
  `name` varchar(255) NOT NULL,
  `access_level_id` int(1) DEFAULT NULL,
  `department_id` int(2) DEFAULT NULL,
  `card_number` bigint(13) NOT NULL
)

--
-- Dumping data for table `user`
--
```

```
INSERT INTO `user` (`id`, `name`, `access_level_id`, `department_id`, `card_number`)
VALUES
(1, 'test_user_admin', 0, NULL, 933583368913);

--
-- Indexes for dumped tables
--

--
-- Indexes for table `access_level`
--
ALTER TABLE `access_level`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `classroom`
--
ALTER TABLE `classroom`
  ADD PRIMARY KEY (`id`),
  ADD KEY `department_id` (`department_id`);

--
-- Indexes for table `department`
--
ALTER TABLE `department`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `operator`
--
ALTER TABLE `operator`
  ADD UNIQUE KEY `username` (`username`),
  ADD KEY `access_level_id` (`access_level_id`);

--
-- Indexes for table `permission`
--
ALTER TABLE `permission`
  ADD PRIMARY KEY (`id`),
  ADD KEY `user_id` (`user_id`),
  ADD KEY `classroom_id` (`classroom_id`);

--
-- Indexes for table `user`
--
ALTER TABLE `user`
  ADD PRIMARY KEY (`id`),
```

```
  ADD KEY `access_level` (`access_level_id`),
  ADD KEY `department` (`department_id`);

--
-- Constraints for dumped tables
--

--
-- Constraints for table `classroom`
--
ALTER TABLE `classroom`
  ADD CONSTRAINT `department_id` FOREIGN KEY (`department_id`) REFERENCES `department`
(`id`);

--
-- Constraints for table `operator`
--
ALTER TABLE `operator`
  ADD CONSTRAINT `access_level_id` FOREIGN KEY (`access_level_id`) REFERENCES
`access_level` (`id`) ON UPDATE CASCADE;

--
-- Constraints for table `permission`
--
ALTER TABLE `permission`
  ADD CONSTRAINT `classroom_id` FOREIGN KEY (`classroom_id`) REFERENCES `classroom`
(`id`),
  ADD CONSTRAINT `user_id` FOREIGN KEY (`user_id`) REFERENCES `user` (`id`);

--
-- Constraints for table `user`
--
ALTER TABLE `user`
  ADD CONSTRAINT `access_level` FOREIGN KEY (`access_level_id`) REFERENCES
`access_level` (`id`),
  ADD CONSTRAINT `department` FOREIGN KEY (`department_id`) REFERENCES `department`
(`id`);
COMMIT;
```

Anexo 3 - Calendário

O calendário Gantt apresenta a lista das tarefas realizadas no desenvolvimento da solução proposta. Foi usado um código de cores para identificar as categorias de cada tarefa tornando melhor a sua interpretação. Seguindo as seguintes regras:

Categoria	Cor
Pesquisa	Vermelho
Desenvolvimento	Rosa
Relatório	Verde
Diagramas	Azul
Código	Azul Claro
Maquete	Laranja
Testes	Amarelo

Tabela 6. - Código de cores

ID	Name	Start	Finish
1	**Fase 1 - Relatório intercalar 1º semestre**	2023-10-12	2023-11-22
4	Pesquisa	2023-10-12	2023-10-24
7	Desenvolvimento da solução proposta	2023-10-25	2023-11-06
8	Analise de recursos necessários	2023-11-05	2023-11-09
2	Construção do relatório	2023-10-22	2023-11-12
9	Elaboração de um questionário	2023-11-04	2023-11-10
10	aplicação do questionário	2023-11-09	2023-11-13
3	Entrega intercalar 1º Semestre	2023-11-22	2023-11-22
5	**Fase 2 - Relatório intermédio**	2023-11-20	2024-01-19
33	Definição de requisitos do sistema	2023-11-20	2023-12-04
18	Desenvolvimento de Diagramas	2023-12-01	2024-01-01
19	BPMN	2023-12-01	2023-12-13
20	UML - Use Case	2023-12-08	2023-12-20
21	Sequência	2023-12-12	2023-12-24
24	Sys Arq (FMC)	2023-12-17	2023-12-29
22	Desenvilvimento de modelo 3D	2023-12-13	2024-01-01
23	Desenvolvimento maquete V1	2023-12-17	2024-01-05
32	Aquisição de componentes	2024-01-04	2024-01-07
26	Desenvilvimento de modelo BD	2024-01-06	2024-01-10

6	Atualização do relatório	2024-01-11	2024-01-17
17	Entrega Intermédia	2024-01-19	2024-01-19
11	**Fase 3 - Relatório Intercalar 2º Semestre**	2024-01-29	2024-04-15
25	Desenvolvimento de código base para o sistema	2024-01-29	2024-02-19
29	Desenvolvimento de BD	2024-02-17	2024-02-26
27	Mock Up Back office	2024-02-10	2024-03-05
28	Implementação de componentes na maquete	2024-03-08	2024-03-21
30	Testes	2024-03-18	2024-03-31
16	Atualização do Relatório	2024-04-02	2024-04-11
13	Entrega Intercalar 2º Semestre	2024-04-15	2024-04-15
12	**Fase 4 - Entrega Final**	2024-04-19	2024-06-28
35	Implementação de funcionalidades do sistema	2024-04-19	2024-05-09
31	Desenvolvimento de maquete final	2024-05-01	2024-05-20
34	Implementação de Back Office	2024-05-09	2024-06-02
36	Testes - Back Office	2024-05-28	2024-06-21
37	Testes - Maquete Final	2024-05-28	2024-06-21
15	Atualização do Relatório	2024-06-15	2024-06-24
14	Entrega Final	2024-06-28	2024-06-28

Tabela 7. - Calendário

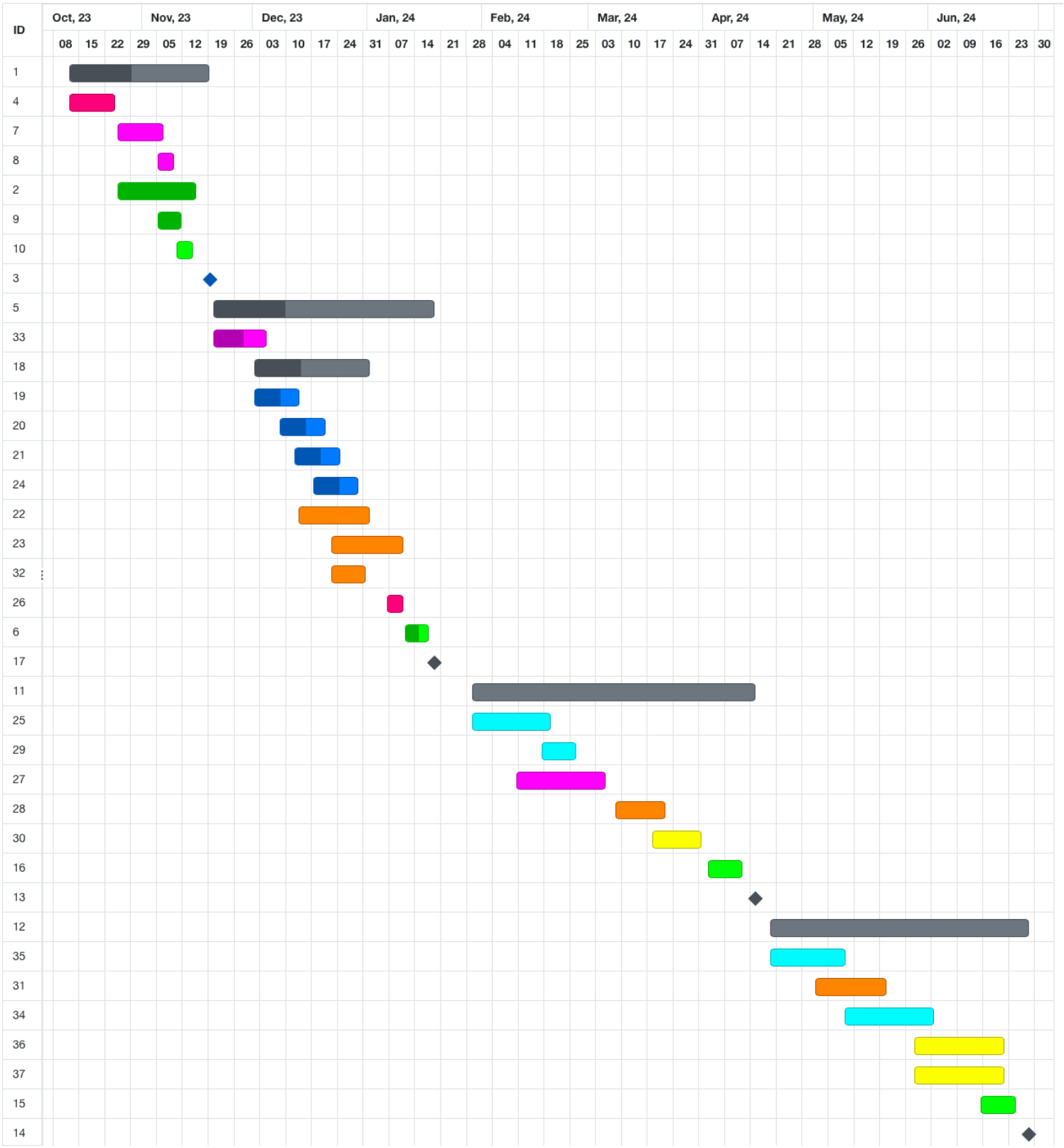


Figura 19. - Calendário Gantt

Glossário

LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
TFC	Trabalho Final de Curso
Back-Office	centro de um software que visa às actividades empresariais, sendo invisível aos utilizadores, mas que disponibiliza, reencaminha e coordena algumas tarefas, para que os utilizadores finais possam efectuar os interesses afins.
RFID	Radio-frequency identification
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
JS	JavaScript
UPS	uninterruptible power supply
BD	Base de Dados
AP	Access Point
UDP	User Datagram Protocol (Protocolo de datagrama do usuário, é um protocolo simples da camada de transporte
GPIO	general-purpose input/output
Hard coding	prática de desenvolvimento de software de embutir dados diretamente no código fonte de um programa ou outro objeto executável, ao invés de obter os dados de fontes externas ou gerá-los em tempo de execução.
E-Ink	tecnologia de display de papel eletrónico