



UNIVERSIDADE
LUSÓFONA

ULHT PresCheck (Back-end)

Trabalho Final de curso

Relatório Final

Alexandre Nunes Garcia - 22004278

Diogo Silva - 22103439

Orientador: Professor João Pedro Leal Abalada de Matos Carvalho

Trabalho Final de Curso | LEI | 21/07/2023

www.ulusofona.pt

Direitos de cópia

ULHT PresCheck (Back-end), Copyright de Alexandre Nunes Garcia e Diogo Silva, ULHT.

A Escola de Comunicação, Arquitetura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) 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 objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Resumo

O presente documento resulta de um projeto a ser desenvolvido no contexto de trabalho final de curso dos autores, que tem por título 'ULHT PresCheck'. Este projeto incide-se no desenvolvimento de uma plataforma web que permita aos seus utilizadores, nomeadamente, os professores da Universidade Lusófona de Humanidades e Tecnologias (ULHT), validar e consultar as presenças das aulas que foram lecionadas de forma simplista para que o período durante a aula destinado à confirmação de presenças, seja por via verbal ou escrita pelo professor e/ou pelos alunos, seja eficientemente otimizado. Estes tipos de plataformas enquadram-se numa categoria de ferramentas de gestão de assiduidade.

Deste modo, o que é uma plataforma de gestão de assiduidade? Apesar do seu nome ser maioritariamente autoexplicativo, algumas pessoas podem não perceber para que a plataforma serve ou como se poderá aproveitar da melhor forma as suas funcionalidades. Apesar deste tipo de soluções não serem palpáveis, são necessários agentes que iniciam ações regulatórias e processos de validação para confirmar a identidade da pessoa e manter a integridade e coerência das presenças, podendo serem considerados dados sensíveis e, por isso, exigem uma atenção própria no processamento desta informação.

Diferentes entidades podem ter regulamentos de assiduidade diferentes e, no caso de entidades académicas, é uma atividade constante, realizada milhares de vezes ao dia, o que exige uma grande capacidade de sincronização e uma arquitetura bastante estruturada e facilmente escalável. Para estas necessidades, cada vez mais presentes com a evolução do mundo tecnológico e para a capacidade de resposta aos seus utilizadores, existem, atualmente, diversos tipos de software, mais, ou menos, específicos, no mercado.

Partindo deste conhecimento, são detetadas algumas falhas e problemas nestes softwares, que na sua grande maioria, não são apropriadas às necessidades reais das entidades de ensino superior, sendo que muitas das vezes, as próprias entidades desenvolvem sistemas que possam dar resposta internamente. O produto final deste projeto visa a ser uma aplicação web de custo baixo a moderado, que permitirá à universidade organizar, estruturar e, se houver necessidade, implementar em outras entidades de ensino superior dentro do Grupo Lusófona.

O sistema desenvolvido irá ter como base a facilitação da marcação de presenças dos alunos em aula e a consulta das mesmas de forma visual e perceptível para acompanhar o percurso de assiduidade às unidades curriculares que, para fins estatísticos e avaliativos, são necessários para os professores para que possam ter uma melhor gestão e eficiência na tomada de decisões da avaliação da componente de participação, fortemente presente nas fichas das unidades curriculares (FUC). Esta plataforma assenta numa solução partilhada entre vários componentes integrados para que seja possível tirar o maior proveito das atuais ofertas tecnológicas do mercado, sendo mais interativo a sua aplicação e usabilidade.

Palavras-chaves: Assiduidade, Gestão de Presenças, Eficiência, Organização, Aplicação Web, Python, Presenças, Estatísticas.

Abstract

The present document is the result of a project to be developed within the context of the authors' final course work, titled 'ULHT PresCheck.' This project focuses on the development of a web platform that allows its users, specifically the professors of the Universidade Lusófona de Humanidades e Tecnologias (ULHT), to validate and consult class attendance in a simplified manner. The aim is to optimize the time allocated during the class for attendance confirmation, whether through verbal or written communication by the professor and/or students. Such platforms fall into the category of attendance management tools.

Thus, what is an attendance management platform? Although its name is mostly self-explanatory, some individuals may not understand the purpose of the platform or how to best utilize its functionalities. Despite these types of solutions being intangible, there is a need for agents to initiate regulatory actions and validation processes to confirm individuals' identities and maintain the integrity and coherence of attendance records, which can be considered sensitive data. Therefore, special attention must be given to the processing of this information.

Different entities may have different attendance regulations, and in the case of academic institutions, attendance management is a constant activity performed thousands of times each day. This demands significant synchronization capabilities and a well-structured and easily scalable architecture. To meet these needs, increasingly present in the evolving technological world and to effectively serve users, there are currently various types of software available in the market, some more specific than others.

Based on this understanding, certain flaws and issues are identified in these software solutions, which, for the most part, are not suitable for the real needs of higher education institutions. Often, these institutions develop their own systems to address these shortcomings internally. The final product of this project aims to be a low-to-moderate-cost web application that will allow the university to organize, structure, and, if necessary, implement it in other higher education entities within the Lusófona Group.

The developed system will be based on facilitating the marking of students' attendance in class and providing a visually accessible means of consulting the attendance records. This will enable tracking the attendance progress in the curricular units, which is necessary for statistical and evaluative purposes. This information is crucial for professors to better manage and make decisions regarding the evaluation of the participation component, which is strongly emphasized in the curricular unit sheets (FUC). The platform is built on a shared solution among multiple integrated components to maximize the benefits of current technological offerings in the market, ensuring a more interactive application and enhanced usability.

Keywords: Attendance, Attendance Management, Efficiency, Organization, Web Application, Python, Presence, Statistics.

Índice

1	Identificação do Problema	1
2	Viabilidade e Pertinência	2
3	Benchmarking	3
4	Engenharia	4
4.1	Levantamento e análise dos requisitos	4
4.1.1	Requisitos Não Funcionais	4
4.1.2	Requisitos Funcionais	5
4.2	Modelos relevantes	7
4.2.1	Diagrama de Entidade-Relação da Base de Dados	7
4.3	Estrutura em árvore da solução	8
5	Solução Desenvolvida	9
5.1	Introdução	9
5.2	Arquitetura	10
5.3	Tecnologias e Ferramentas Utilizadas	11
5.4	Implementação	11
5.5	Abrangência	12
6	Método e Planeamento	13
7	Resultados	15
8	Conclusão e trabalhos futuros	16
•	Bibliografia	17
•	Anexo 1 – Questionário de viabilidade	18
•	Glossário	19

Lista de Figuras

Figura 1 - Modelo de Entidade-Relação da base de dados da solução.	7
Figura 2 - Modelo de Entidade-Relação da base de dados da solução final.	7
Figura 3 - Arquitetura global da solução final. Adaptado a partir de React [REACT], Flask [FLASK] e SQLite [SQLITE].	10
Figura 4 - Arquitetura para a leitura de dispositivo do aluno. Adaptado a partir de Flask [FLASK] e Arduino [ARDUINO].	10
Figura 5 - Calendário em formato Gantt do planeamento do projeto.	13

Lista de Tabelas

Tabela 1 – Tabela de conclusão dos requisitos não funcionais face às entregas	14
Tabela 2 – Tabela de conclusão dos requisitos funcionais face às entregas	14
Tabela 3 – Tabela de referência de preços para os componentes do arduíno e controladores. Preços vistos na data da entrega na loja de eletrónica Mauser [MAUSER].	15

1 Identificação do Problema

Com todo o avanço tecnológico e a transformação digital que ocorre nos dias de hoje, ainda existem atividades de certas entidades que não estão totalmente exploradas devido a atrasos ou prioridades administrativas. Atualmente, no Grupo Lusófona, e mais especificamente na Universidade Lusófona de Humanidades e Tecnologias (ULHT), sendo uma das universidades privadas mais conceituadas nacionalmente, possui um método de marcação de presenças e assiduidade bastante manual, ou seja, não existe nenhum sistema central que seja igual em todas as cadeiras, o que leva os professores a perderem grande parte do seu tempo a fazer a chamada ou à espera que todos os alunos assinem uma folha de papel para que mais tarde tenha que inserir nas suas grelhas manualmente.

Por causa desta situação, muitos professores optam por não realizar a chamada nem marcar as presenças, escolhendo soluções alternativas, tais como anotar apenas o número de alunos presentes na sala, ou nem fazem este processo de todo, visto que em turmas grandes, qualquer método é impraticável e consome mais tempo de aula do que é o aceitável. Para além disso, cada professor tem o seu próprio meio e ferramentas para dar resposta a este problema, e não segue uma estrutura padronizada que possa tirar um aproveitamento crescente da informação que é inserida.

Nos dias que correm, os alunos são cada vez mais mantidos na ignorância em termos da sua assiduidade. Apesar de terem a sua própria responsabilidade, muitos não sabem se as faltas são marcadas e nem os professores dizem. Podemos perguntar a um grande universo de alunos se tem conhecimento da gestão de assiduidade da Universidade Lusófona e muitos nem têm conhecimento que existe uma página de faltas na secretaria virtual (NetPA) da faculdade, tampouco o seu histórico e percurso do ano letivo corrente e anteriores.

O mercado destas soluções evoluiu, mas muito poucas empresas são contratadas para desenvolver este tipo de sistemas, inevitáveis e complexos, para as instituições de ensino públicas e privadas, o que deixa um grande buraco em aberto, forçando as próprias instituições a desenvolver este tipo de soluções, pois são sistemas muito personalizados e acaba por ser necessário adaptar à realidade da faculdade.

2 Viabilidade e Pertinência

O projeto em questão tem capacidade para garantir a viabilidade de desenvolvimento para além do âmbito do trabalho académico, visto que existe espaço para aumentar a escala de aplicação a qualquer instituição de ensino, seja superior ou secundário, pública ou privada. Nesta fase inicial, o projeto será focado num universo reduzido e não existirá integração ou comunicação com a plataforma NetPA, não obstante que em futuros incrementos este modelo de software pode ser implementado em todos os departamentos da ULHT e do Grupo Lusófona.

Como referido anteriormente, nesta fase a plataforma tem como principal objetivo responder às necessidades de assiduidade dos alunos nas instituições de ensino, reduzindo assim o esforço e tempo despendido em interações entre os professores e alunos durante o processo de marcação de presenças em sala de aulas ou posteriormente em momentos de avaliação. Com isto, os professores poderão aplicar este tempo otimizado para obter o melhor aproveitamento possível dos conteúdos programados para a aula a lecionar e ajudar os alunos a terem uma compreensão ideal da gestão e responsabilidade da assiduidade às aulas.

A metodologia aplicada no desenvolvimento deste projeto permite que o mesmo seja escalável e adaptável a qualquer tipo de instituição de ensino, permitindo assim um uso mais universal. A utilização de um servidor físico local pode também ser um conceito a ser melhorado, existindo a possibilidade de exportar todo o conteúdo para serviços ‘cloud’ e melhorar a eficiência da arquitetura quase imediatamente.

Realizou-se um questionário anónimo, com o auxílio da plataforma Microsoft Forms, aos professores do DEISI sobre as atuais limitações deste processo de presenças [Anexo 1 - Questionário]. Foram obtidas 17 respostas a este questionário e com os resultados, podemos perceber o défice de eficiência deste processo e é de notar a variedade de ferramentas que os docentes utilizam como formas de mitigar este constrangimento.

A proposta inicial deste projeto manteve-se durante todo o processo de desenvolvimento, há exceção do acesso e consulta por parte dos alunos, onde apenas os professores e pessoal administrativo poderá ter acesso à plataforma para gestão de alunos, turmas e outras entidades necessárias para manter a integridade e atualização da informação.

3 Benchmarking

No mercado empresarial existem, como se sabe, numerosos sistemas de informação de gestão de assiduidade. Estes são, em geral, adaptáveis às estruturas empresariais, tradicionalmente organizadas em departamentos, mas raramente são aplicáveis a instituições de ensino. Exemplos de sistemas de controlo de assiduidade de alunos são:

- SiGeFA - Sistema de Gestão de Frequências de Alunos [Universidade Aberta, CISUC - Centro de Informática e Sistemas da Universidade de Coimbra]

“(..)Sistema de Gestão de Frequências de Alunos (SiGeFA) composto por uma aplicação móvel, uma aplicação web e aplicação backend que visa monitorizar a assiduidade dos alunos por meio de marcação de faltas dos alunos ausentes e de envio automático de notificações aos pais e encarregados de educação cujos alunos estiverem ausentes durante as aulas.” (SiGeFA, 2016)

- Muffin [Subotica Tech, Department of Informatics, Subotica, Serbia]

“Muffin consists of a mobile application that students have on their personal mobile devices, an Arduino Uno board with Bluetooth module, and a desktop application. The basic idea is that when students want to log in their attendance, all they need to do is send some log in data to the Arduino board, which is then forwarded to the application on the desktop PC that is directly connected to the board.” (Muffin, 2016)

A estrutura de uma instituição de ensino superior difere de uma estrutura empresarial, e na maioria dos casos, a sua estrutura interna difere de instituição para instituição. Atualmente existem alguns, ainda que poucos se considerarmos a evolução do ensino superior, sistemas de gestão de assiduidade direcionados para o ensino superior. Estes sistemas, são essencialmente direcionados para a gestão corrente de secretarias, como no caso da ULHT, o NetPA.

Em que o *ULHT PresCheck* difere destes sistemas? Em primeiro lugar é necessário referir que a ULHT, apesar de não ter um sistema de assiduidades integrado, tem já muita informação em suporte informático e em formato importável para uma base de dados relacional. Em segundo lugar, a plataforma a desenvolver deve integrar toda a informação relativamente a presenças dos alunos, dado que, em muitos casos, é informação oficial. Finalmente existem as vertentes pedagógica e tecnológica que este projeto se propõe a contemplar e que não está presente nos outros projetos referidos. Realça-se que o pretendido neste projeto se assemelha a uma conjunção entre os dois projetos referidos acima, mas utilizando tecnologias mais desenvolvidas e otimizadas para o efeito.

Existem ainda artigos e autores que sugerem o uso de reconhecimento facial dos alunos através de inteligência artificial. Neste tipo de abordagem o professor tira fotografia(s) da sala de aula incluindo a face de todos os alunos e a partir destas fotografias o sistema identifica os alunos possibilitando depois o registo da sua assiduidade. Contudo este tipo de solução pode ter alguma imprecisão, dado que podem não ser identificados todos os alunos e requer treino dos algoritmos com fotografias individuais dos alunos, e nos dias atuais, existe uma forte presença do Regulamento Geral de Proteção de Dados (RGPD) em vigor devido à salvaguarda consentida dos dados biométricos de qualquer pessoa.

4 Engenharia

Este projeto recai exatamente sobre o problema identificado acima, e tem como objetivo colmatar algumas destas dificuldades das instituições de ensino, possibilitando assim uma melhor gestão de assiduidades, de modo que o foco seja a genialidade na produção de soluções aos problemas.

Não existe uma fórmula que garanta o sucesso desta adaptação uma vez que cada professor tem uma filosofia e metodologia diferente, assim é necessário que a solução encontrada para estes problemas seja o mais abrangente possível, mas em simultâneo que possa ser detalhada e personalizada para as necessidades específicas do negócio.

4.1 Levantamento e análise dos requisitos

Atualmente, todos os sistemas informáticos têm como objetivo responder a objetivos e/ou necessidades dos utilizadores, sendo estes utilizadores, singulares ou organizações. De forma que estes sistemas tenham sucesso, é necessário que seja estabelecida uma ponte entre a equipa de desenvolvimento e as organizações/utilizadores. Deste modo, existe uma enorme importância para a correção e precisa definição de requisitos de sistema, uma vez que, são estes que funcionam como guia para gestores de projeto, programadores, técnicos de sistemas e elementos de qualidade, durante o desenrolar do projeto.

Assim, numa fase inicial deste projeto, procurámos que requisitos haveriam de ser implementados e cumpridos para que esta respondesse às necessidades dos nossos utilizadores e cumprisse na íntegra a visão que tínhamos projetado sobre o seu estado final. Foram então estipulados requisitos em duas formas: requisitos não funcionais e funcionais, identificados com uma importância de escala de tamanhos¹ e por esforço esperado².

4.1.1 Requisitos Não Funcionais

RNF1. A aplicação deverá facilitar todo o processo de marcação de presenças durante uma aula, respeitando as leis e regras impostas para este efeito. **[XL – 34 pontos]**

RNF2. A aplicação deverá fornecer métricas e visualizações gráficas de análise de dados, como por exemplo presenças nas unidades curriculares. **[L – 8 pontos]**

RNF3. A aplicação deverá ter embutido uma hash de encriptação ao efetuar login de utilizadores. **[L – 3 pontos]**

RNF4. Para perfis de utilizador com permissões de administrador, o *dashboard* deverá fornecer informações apropriadas às necessidades. **[L – 5 pontos]**

RNF5. Utilização do framework Flask para o desenvolvimento do backend. **[M – 1 ponto]**

RNF6. Utilização da base de dados SQLite para armazenamento de dados. **[M – 1 ponto]**

1. Escala de tamanhos: {XS, S, M, L, XL}.
2. Esforço esperado: pontos por referência de Poker Planning.

RNF7. Comunicação entre dispositivos RFID/NFC e o servidor através de uma rota própria para os Arduinos. **[L – 21 pontos]**

RNF8. A solução deverá ter escalabilidade para suportar grandes números de utilizadores e registos de aulas. **[L – 34 pontos]**

RNF9. É necessário que o sistema tenha um desempenho e velocidade aceitáveis na resposta aos pedidos dos utilizadores. **[M – 34 pontos]**

RNF10. O sistema deve ter uma disponibilidade de 99,5% do tempo online. **[M – 34 pontos]**

RNF11. O sistema deve ser testado antes do lançamento para garantir a qualidade. **[M – 21 pontos]**

RNF12. O sistema deve garantir a segurança da informação que recolhe sobre os utilizadores. **[L – 55 pontos]**

4.1.2 Requisitos Funcionais

RF1. Utilizadores com permissão de administração devem poder criar, modificar e eliminar os acessos dos professores à plataforma. **[XL – 8 pontos]**

RF2. Utilizadores com permissão de administração deverão conseguir associar unidades curriculares a pelo menos um professor. **[XL – 5 pontos]**

RF3. Deverá ser possível editar qualquer tipo de entidade, caso haja permissão. **[L – 8 pontos]**

RF4. Deverá ser possível remover qualquer tipo de entidade, caso haja permissão. **[M – 8 pontos]**

RF5. Para utilização da aplicação, é necessário ter um login de utilizador válido e ativo. **[XL – 5 pontos]**

RF6. Após o login bem-sucedido, o utilizador deverá ser redirecionado para o *dashboard* pessoal. **[L – 3 pontos]**

RF7. O sistema deve permitir que os professores possam registar os alunos na plataforma, fornecendo o número do aluno e o UID do dispositivo RFID/NFC, se necessário. **[XL – 13 pontos]**

RF8. O sistema deve ser capaz de registar a presença dos alunos nas aulas, utilizando Arduinos. **[XL – 34 pontos]**

RF9. O sistema deve ser capaz de registar a presença dos alunos nas aulas por introdução do número manualmente pelo professor. **[M – 13 pontos]**

RF10. O sistema deve permitir aos professores editar informações dos alunos, incluindo número do aluno, nome e UID do dispositivo RFID/NFC associado ao aluno. **[L – 13 pontos]**

RF11. O sistema deve permitir aos professores fornecer a lista de alunos presentes em uma aula específica no tempo. **[XL – 8 pontos]**

RF12. O sistema deve permitir que os professores possam fornecer estatísticas de presença das aulas, por dia e/ou por número de alunos. **[L – 5 pontos]**

RF13. O sistema deve permitir que os professores possam fornecer estatísticas de presença das aulas, por mês e/ou por número de alunos. **[L – 5 pontos]**

RF14. O sistema deve permitir que os professores possam fornecer estatísticas de presença das aulas, por semestre e/ou por número de alunos. **[L – 5 pontos]**

RF15. O sistema deve permitir aos professores transferir a lista de presenças em formato CSV. **[M – 5 pontos]**

Para a entrega final do projeto, todos os requisitos foram cumpridos, há exceção dos seguintes requisitos, que foram alterados para compreender e implementar o módulo de gestão e estatísticas de melhor forma:

- **RF7 e RF10** – Os professores só conseguem ver estatísticas e marcar a presença manualmente do aluno. Toda a gestão de dispositivos (criação, modificação e eliminação) associados aos alunos ficou associada ao administrador do sistema.
- **RF12, RF13 e RF14** – Estes requisitos foram alterados para englobar os dados mais uniformemente. Sendo assim, a errata dos mesmos ficou:
 - **RF12.** O sistema deve permitir que os professores possam ver estatísticas de média, mediana e número de presenças por unidade curricular associada. **[L – 5 pontos]**
 - **RF13.** O sistema deve permitir que os professores possam ver estatísticas de histórico das presenças do aluno relativamente à unidade curricular selecionada. **[L – 5 pontos]**
 - **RF14.** O sistema deve permitir que os professores possam ver estatísticas sobre a média de atrasos numa unidade curricular, definindo um tempo de tolerância. **[L – 5 pontos]**

4.2 Modelos relevantes

4.2.1 Diagrama de Entidade-Relação da Base de Dados

À data de 27 de janeiro de 2023, a solução apresenta-se ainda rudimentar nas funcionalidades sendo que esta fase foi focada na criação da base de dados e respetivas relações [Figura 1].

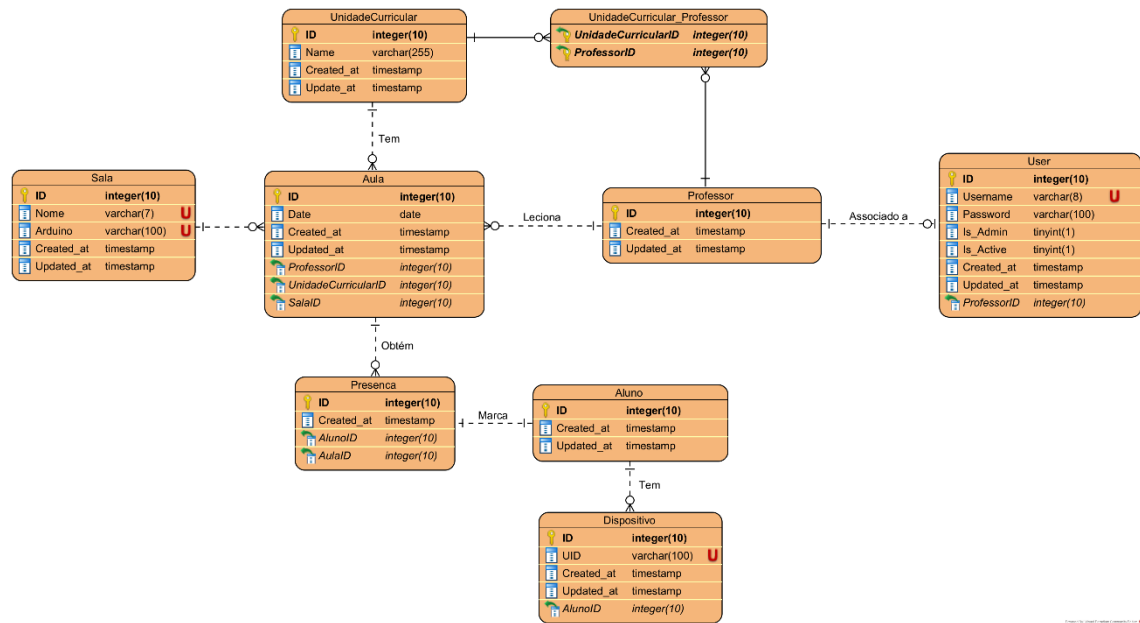


Figura 1 - Modelo de Entidade-Relação da base de dados da solução.

Durante o período de desenvolvimento até à entrega final, a base de dados sofreu alterações para abranger novas entidades, necessárias para uma melhor organização, eficiência e implementação de estatísticas [Figura 2].

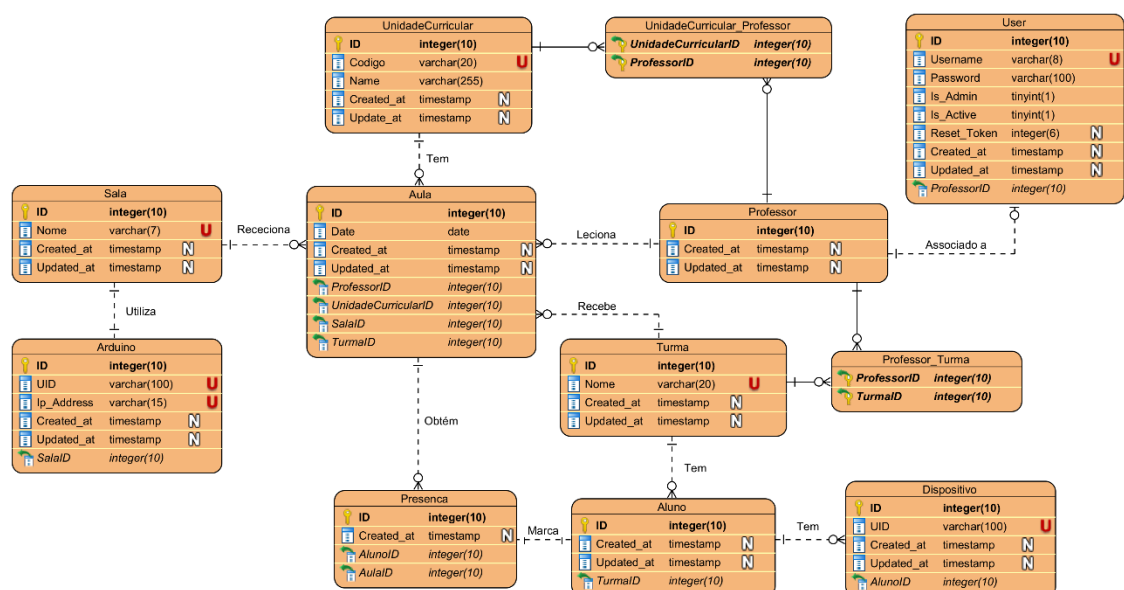


Figura 2 - Modelo de Entidade-Relação da base de dados da solução final.

Para a tabela de “User”, foi adicionada uma nova coluna chamada “Reset-Token” para que seja possível armazenar o código de recuperação da senha do professor. Este campo é limpo quando uma senha é definida.

De forma a conseguir fazer consultas e obter estatísticas das presenças por turno/turma, foram adicionadas as tabelas “Turma” e “Professor_Turma”, onde o aluno está associado apenas a uma turma e o professor pode dar aulas a várias turmas. Nas aulas, também é registo a que turma foi dada a aula pelo professor.

Para incluir as novas informações sobre os controladores e arduinos, a coluna “Arduíno” foi retirada da tabela de “Sala” e foi adicionada uma nova tabela específica para a entidade “Arduíno”, sendo que um arduíno está associado apenas a uma sala.

Dado que a ULHT identifica as suas unidades curriculares por um código com caracteres (e.g “Trabalho Final de Curso” de LEI tem o código ULHT260-5140) [TFC], a tabela “Unidade” foi alterada para incluir este código e permitir que a unidade seja encontrada facilmente na plataforma.

4.3 Estrutura em árvore da solução

Back-end:

```
/
├── .flaskenv (Variáveis de ambiente)
├── app.py (Inicialização do servidor Flask)
├── admin.py (Rotas para o módulo de administração/gestão)
├── auth.py (Rotas para o módulo de autenticação e gestão da conta)
├── config.py (Configurações do servidor Flask)
├── main.py (Rotas para o módulo de registo, gestão e encerramento de aulas)
├── stats.py (Rotas para o módulo de estatísticas)
├── models.py (Definição das entidades da base de dados em ORM)
├── requirements.txt (Dependências do servidor Flask para a sua execução)
├── ulht-prescheck.db (Base de dados da plataforma)
├── templates (Pasta para salvaguarda dos templates de e-mail)
│   ├── new_account.html (Template do e-mail de criação de acesso)
│   └── reset_password.html (Template do e-mail de recuperação de senha)
```

Controlador do arduino:

```
/
├── .flaskenv (Variáveis de ambiente)
├── app.py (Inicialização do servidor Flask e rotas de controlo da leitura do arduino)
├── leitor.py (Ficheiro de leitura do arduino)
├── requirements.txt (Dependências do servidor Flask para a sua execução)
```


5 Solução Desenvolvida

5.1 Introdução

A proposta é desenvolver uma aplicação web acessível em termos de custo, com uma interface fácil de utilizar. Essa aplicação funcionará como uma plataforma de gestão de presenças para a Universidade Lusófona, permitindo a leitura das presenças dos alunos de forma eficiente. Para alcançar esse objetivo, será realizada uma integração com componentes externos, especificamente dispositivos arduino, que serão associados a cada aluno. Esses dispositivos permitirão que os alunos marquem as suas presenças nas aulas da universidade. Para garantir a autenticidade e evitar fraudes na identificação dos alunos, a implementação contará com medidas de segurança. Será programado um tempo de espera no dispositivo arduino após o aluno marcar sua presença. Esse tempo permitirá ao professor confirmar a identidade do aluno que utilizou o dispositivo, aumentando a confiabilidade do sistema.

Este relatório abordará principalmente o desenvolvimento do back-end da plataforma, que é a parte responsável pelo processamento e armazenamento dos dados. Será detalhado como os dados das presenças serão geridos, processados e armazenados na base de dados, bem como a integração com os dispositivos arduino será implementada. É importante ressaltar que o foco principal será na funcionalidade e eficiência do back-end.

A aplicação como um todo não foi planeada para integrar-se diretamente com outros sistemas da Universidade Lusófona, como o NetPA ou a aplicação móvel "Ensino Lusófona". Entretanto, é possível considerar essa possibilidade como um desenvolvimento futuro, caso seja necessário. Por enquanto, o objetivo é criar uma solução independente e autónoma para a gestão de presenças dos alunos, focada na sua eficácia e facilidade de uso.

Além disso, é essencial considerar medidas de segurança adicionais para proteger os dados dos alunos e garantir a privacidade de suas informações pessoais, conforme as regulamentações aplicáveis, como o Regulamento Geral de Proteção de Dados (RGPD) ou outras leis de proteção de dados vigentes onde a universidade está localizada.

Link para o repositório github:

- Servidor back-end: <https://github.com/DEISI-ULHT-TFC-2022-23/TFC-DEISI308-ULHT-PresCheck-Backend>
- Servidor para controlador e código fonte do arduino: <https://github.com/AlexandreGarcia-a22004278/ulhtprescheck-controller>

5.2 Arquitetura

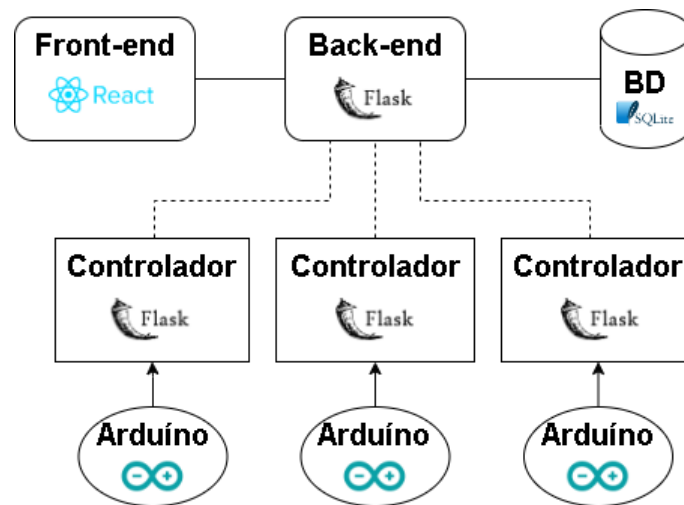


Figura 3 - Arquitetura global da solução final. Adaptado a partir de React [REACT], Flask [FLASK] e SQLite [SQLITE].

Conforme a figura acima [Figura 3], O servidor back-end é o sistema central da solução, servindo como uma API REST (Application Programming Interface Representational State Transfer) para receber pedidos e enviar respostas com dados e informações para o servidor front-end. A comunicação entre estes dois servidores, front-end e back-end é feita pela formatação JSON (JavaScript Object Notation) [JSON] e encriptada utilizando a tecnologia JWT (JSON Web Tokens) [JWT], que garante a autenticação e integridade dos dados pedidos pelo front-end e enviados pelo back-end.

Para servir o front-end com dados coerentes e dinâmicos, o back-end mantém uma comunicação direta com a base de dados, formatada em ORM (Object-Relational Mapping), que possui uma linguagem própria e que permite isolar a responsabilidade de toda e qualquer ação necessária na base de dados, há exceção de consultas, nas próprias classes das entidades.

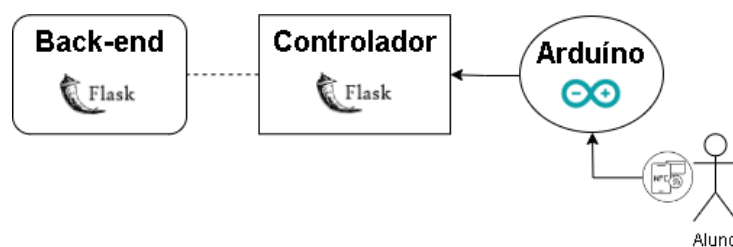


Figura 4 - Arquitetura para a leitura de dispositivo do aluno. Adaptado a partir de Flask [FLASK] e Arduino [ARDUINO].

De forma a cumprir o objetivo deste projeto, otimizar o processo de marcação de presenças através de dispositivos externos, foi adicionado um servidor adicional, denominado de “controlador”, como demonstra a figura acima [Figura 4], com a função de API exclusiva para a comunicação com o back-end em formato JSON [JSON]. Este servidor intermédio pode ser instalado em qualquer computador/microcomputador (e.g Raspberry Pie) e deve estar ligado fisicamente ao arduíno, por comunicação em série entre estes. Desta forma, é possível ler os dados recolhidos do dispositivo do aluno e seguidamente remeter os mesmos, devidamente encriptados através da tecnologia JWT [JWT], para o back-end.

5.3 Tecnologias e Ferramentas Utilizadas

Para o back-end e controladores, é utilizada a tecnologia *Flask* [FLASK], uma framework para desenvolvimento web baseada na linguagem de programação *Python* [PYTHON]. Esta framework destaca-se entre as restantes frameworks para o mesmo efeito devido à sua capacidade de resposta, integração simples e facilmente escalável, permitindo instalar apenas as dependências necessárias para o seu funcionamento. Toda a comunicação com estes sistemas é realizada no formato JSON [JSON] e encriptada através da tecnologia JWT [JWT].

Na gestão de dados, utilizou-se a tecnologia *SQLite* [SQLITE], um motor de gestão de base de dados leve, de confiança e altamente eficiente, perfeito para implementar em projetos que exigem grandes fluxos de transmissão e salvaguarda de dados, sendo a escolha de eleição da comunidade de Python [PYTHON], pois está presente nas bibliotecas raiz da linguagem, o que permite evitar a instalação de dependências para iniciar e executar ações na base de dados. Para fazer a interligação entre o motor de base de dados e o Flask, optou-se por usar a biblioteca do Python [PYTHON] chamada *SQLAlchemy* [SQLALCHEMY], onde podemos criar as entidades da base de dados e as suas relações através de classes de objetos e traduzindo para SQL (Structured Query Language) [SQL] utilizando o método ORM (Object-Relational Mapping), garantindo uma proteção adicional contra ataques à base de dados (e.g. SQLInjection).

Para realizar a marcação de presenças automatizada do aluno, decidimos recorrer ao microcontrolador *Arduino* [ARDUINO] e um módulo chamado *MFRC522* [MFRC522] para detetar e ler tags RFID/NFC (Radio Frequency Identification/Near Field Communication) [RFID], duas tecnologias atuais que estão presentes em quase todos os itens que acompanham uma pessoa no dia-a-dia (telemóvel, carteira, chaves, entre outros). Este microcontrolador depende sempre de alguma fonte de energia, e para esta arquitetura, o mesmo estará ligado constantemente a um microcomputador por ligação USB (Universal Serial Bus).

5.4 Implementação

De acordo com os requisitos elicitados no ponto 4 e com as normas de segurança dos dados sensíveis, o servidor back-end faz a encriptação destes ao salvar na base de dados e envia e-mails ao professor com uma senha auto-gerada pelo sistema, no ato da criação de conta pelo administrador, ou um código para a recuperação da sua senha, caso se tenha esquecido desta. Os administradores do sistema não podem alterar a senha nem iniciar um processo de recuperação para os professores.

No caso do registo e gestão de aulas a decorrer, o servidor possui estruturas de dados capazes de organizar e conter a Informação necessária para a marcação das presenças e outras informações da aula que são necessárias para identificar a aula e garantir a integridade da Informação. Quando um professor inicia um novo registo de uma aula, o servidor back-end recolhe as informações do formulário preenchido e envia a comunicação para o controlador que está associado à sala escolhida para iniciar o processo de leitura dos dispositivos dos alunos. O arduino, por si, deteta um dispositivo em aproximação, lê o seu UID e remete o mesmo ao controlador em tempo real, que por sua vez envia o UID em conjunto com a sua identificação para o back-end, onde são feitas as restantes validações e por fim é adicionado à lista de alunos presentes na aula. Dada a aula por encerrada, todas as informações guardadas

temporariamente nas estruturas próprias são salvaguardadas na base de dados por rotinas, ativadas no ato de encerramento da aula.

Para a parte das estatísticas, quando o front-end faz o pedido ao servidor back-end, o mesmo realiza a query e resolve a computação da média e outros dados estatísticos do lado do motor de base de dados para obter a maior eficiência possível. A fórmula aplicada para a média das presenças, filtrada por aluno, turma ou unidade curricular segue a base matemática de:

$$\frac{\sum \text{presenças por aula}}{\text{total de aulas}}$$

Os administradores podem aceder a páginas privadas para manter a boa gestão de todas as entidades, tais como unidades, alunos, salas, turmas, criar contas de utilizador para os professores, associar unidades e turmas ao professor, ter um histórico completo de todas as aulas dadas e gerir as aulas que estão a decorrer em tempo real.

Para uma melhor organização e legibilidade do código, o servidor back-end possui rotas separadas por categorias, onde cada categoria tem um ficheiro próprio, conforme a estrutura de ficheiros descrita no ponto 4.3 deste documento.

5.5 Abrangência

Na vertente académica, esta solução vai envolver metodologias já trabalhadas em algumas unidades curriculares lecionadas até à data, sendo estas:

- Programação Web
- Bases de Dados
- Engenharia de Software
- Algoritmia e Estrutura de Dados
- Arquiteturas Avançadas de Computadores

6 Método e Planeamento

Para um melhor entendimento do planeamento durante todo o percurso do projeto, especificamos o desenvolvimento e configuração em várias etapas e tarefas. Devido à questão de os autores serem trabalhadores-estudantes, não foi possível utilizar uma metodologia de planeamento fixa, pois estaríamos a adaptar-nos constantemente na execução e desenvolvimento da solução.

Do início do ano letivo e da apresentação dos temas até à avaliação intermédia do 1º semestre mantivemos um ritmo fixo para elaboração dos diagramas e ao mesmo tempo desenvolver a estrutura do servidor back-end e do arduino. A partir da parte de desenvolvimento, decidimos dar prioridade à realização das restantes cadeiras do semestre e adiar a entrega final para 2ª época devido à falta de tempo e capacidade para desenvolver vários projetos ao mesmo tempo, no entanto, sempre se adotou uma metodologia Agile com reuniões semanais entre as equipas de front-end e back-end e definir o que se iria desenvolver durante aquela semana e em constante verificação dos testes de integração.

Consideramos que o calendário abaixo [Figura 5] relata e descreve o tempo e capacidade de realização das tarefas, sendo que algumas tarefas não foram possíveis de implementar diretamente sem depender de outras funcionalidades, o que levou ao desenvolvimento em paralelo de alguns módulos.

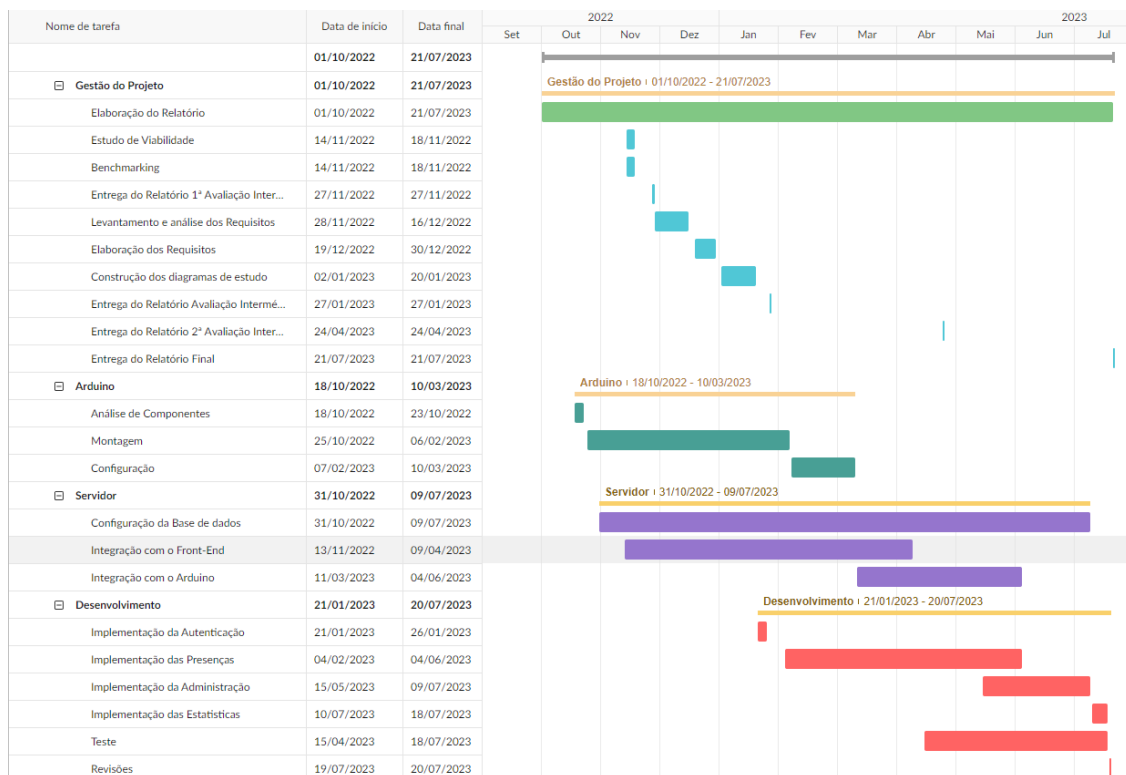


Figura 5 - Calendário em formato Gantt do planeamento do projeto.

O grupo “Gestão de Projeto” possui todas as tarefas de planeamento, análise e “milestones” do projeto. O grupo “Arduino” remete à análise da arquitetura, montagem e programação do arduino-modelo. O grupo “Servidor” possui as tarefas de configuração e

implementação da arquitetura da solução. O grupo “Desenvolvimento” trata-se do desenvolvimento dos módulos da plataforma, os testes internos à API e revisões finais.

De acordo com a calendarização proposta, nas tabelas [Tabela 1 e 2] abaixo é possível ver a data de conclusão dos requisitos funcionais e não funcionais, de acordo com as entregas intercalares e intermédias do projeto.

	Entrega Intercalar 1º Semestre	Entrega Intermédia 1º Semestre	Entrega Intercalar 2º Semestre	Entrega Final 2º Semestre
<i>RNF1</i>				<input checked="" type="checkbox"/>
<i>RNF2</i>				<input checked="" type="checkbox"/>
<i>RNF3</i>			<input checked="" type="checkbox"/>	
<i>RNF4</i>				<input checked="" type="checkbox"/>
<i>RNF5</i>		<input checked="" type="checkbox"/>		
<i>RNF6</i>		<input checked="" type="checkbox"/>		
<i>RNF7</i>			<input checked="" type="checkbox"/>	
<i>RNF8</i>				<input checked="" type="checkbox"/>
<i>RNF9</i>				<input checked="" type="checkbox"/>
<i>RNF10</i>				<input checked="" type="checkbox"/>
<i>RNF11</i>				<input checked="" type="checkbox"/>
<i>RNF12</i>				<input checked="" type="checkbox"/>

Tabela 1 – Tabela de conclusão dos requisitos não funcionais face às entregas.

	Entrega Intercalar 1º Semestre	Entrega Intermédia 1º Semestre	Entrega Intercalar 2º Semestre	Entrega Final 2º Semestre
<i>RF1</i>				<input checked="" type="checkbox"/>
<i>RF2</i>				<input checked="" type="checkbox"/>
<i>RF3</i>				<input checked="" type="checkbox"/>
<i>RF4</i>				<input checked="" type="checkbox"/>
<i>RF5</i>			<input checked="" type="checkbox"/>	
<i>RF6</i>			<input checked="" type="checkbox"/>	
<i>RF7</i>			<input checked="" type="checkbox"/>	
<i>RF8</i>			<input checked="" type="checkbox"/>	
<i>RF9</i>				<input checked="" type="checkbox"/>
<i>RF10</i>			<input checked="" type="checkbox"/>	
<i>RF11</i>				<input checked="" type="checkbox"/>
<i>RF12</i>				<input checked="" type="checkbox"/>
<i>RF13</i>				<input checked="" type="checkbox"/>
<i>RF14</i>				<input checked="" type="checkbox"/>
<i>RF15</i>			<input checked="" type="checkbox"/>	

Tabela 2 – Tabela de conclusão dos requisitos funcionais face às entregas.

7 Resultados

Os resultados deste projeto foram positivos, onde foi possível implementar um serviço web para facilitar o dia dos professores da ULHT poderem acompanhar o ritmo de presenças nas suas aulas e identificar possíveis problemas e/ou riscos que possam prejudicar de alguma forma o acompanhamento pedagógico e avaliativo dos alunos inscritos.

Todos os processos da plataforma foram testados e revistos de forma a encontrar potenciais pontos de falha e otimizar de melhor forma a navegação e usabilidade do serviço. As ações são todas em tempo real, como o envio de e-mails relacionados à conta do professor, da leitura de dispositivos dos alunos pelo arduíno para a marcação da presença ou para o registo do aluno na plataforma.

Os utilizadores com poderes de administração conseguem ter uma visão completa de tudo o que se passa na plataforma, gerir todas as tabelas da base de dados e ter uma visão em tempo real das aulas que estão a decorrer, onde podem ver a lista de alunos presentes, suspender a marcação, encerrar a aula e cancelar a aula, caso se justifique.

Os alunos podem registar-se com qualquer tipo de dispositivo, desde que tenha um UID de RFID ou NFC fixo. Caso não tenha, ou não pretenda associar itens pessoais, como cartões bancários, pode ser atribuído um cartão próprio para a marcação de presenças registado pelo DEISI. Se o aluno desejar eliminar a associação de um dispositivo ou trocar o dispositivo, é possível fazer essa ação pela administração da plataforma.

Dado que a viabilidade deste projeto é pela otimização na marcação de presenças, existem custos imputados para a implementação e configuração do sistema no geral. Aconselha-se a montar um servidor para o back-end com pelo menos 16GB de RAM, 250GB SSD e um CPU com pelo menos 8 cores, tendo em conta que irá receber vários processos e pedidos vindo dos utilizadores da plataforma e será necessário ter um tempo de resposta consideravelmente rápido para uma boa usabilidade. Para além do servidor, existe também custos associados aos arduinos, onde cada esquema para ler um dispositivo pode estar compreendido na seguinte tabela [Tabela 3].

Produto	Preço
Microcontrolador Arduino UNO REV3	29,52€
Módulo NFC MIFARE baseado no NXP MFRC-522 3.3VDC	15,35€
Microcomputador Raspberry Pi 3 Model A+ 1.4Ghz 512Mb com WiFi 2.4/5GHz + Bluetooth 4.2	29,99€
Cartão de proximidade MIFARE RFID 13.56 MHz (10 unidades)	5,80€
Cabos jumper macho-fêmea (10 unidades)	1,00€
Total	81,66€

Tabela 3 – Tabela de referência de preços para os componentes do arduíno e controladores. Preços vistos na data da entrega na loja de eletrónica *Mauser* [MAUSER].

8 Conclusão e trabalhos futuros

O projeto de implementação do serviço web para acompanhamento de presenças e gestão das aulas dos professores da Universidade Lusófona de Humanidades e Tecnologias (ULHT) foi concluído com sucesso, apresentando resultados positivos e contribuindo significativamente para a melhoria do processo pedagógico e avaliativo dos alunos inscritos.

Através da plataforma desenvolvida, os professores obtiveram uma ferramenta eficiente que lhes permitiu acompanhar em tempo real as presenças dos alunos nas suas unidades, identificar potenciais problemas e riscos, e assim, possibilitou-lhes uma atuação mais proativa no acompanhamento académico e na tomada de decisões sobre a unidade curricular.

A realização dos testes e otimizações dos processos da plataforma garantiu um funcionamento estável e seguro, oferecendo uma navegação intuitiva e uma experiência positiva para os utilizadores. Com a incorporação do arduíno, a possibilidade de marcação de presenças e registo dos alunos de forma automatizada e em tempo real trouxe maior agilidade e precisão ao processo, contribuindo para uma gestão mais eficaz das aulas.

Em resumo, o serviço web desenvolvido neste projeto mostrou-se uma solução completa e bem-sucedida para aprimorar o acompanhamento pedagógico e a gestão de aulas na ULHT. A implementação bem-sucedida e os resultados positivos obtidos são um reflexo do empenho e dedicação da equipa envolvida no projeto. Recomenda-se que esta plataforma seja mantida e continuamente atualizada para acompanhar as necessidades em constante evolução da comunidade académica, proporcionando assim um ambiente de ensino mais eficiente e produtivo.

Devido ao tempo de execução e desenvolvimento não ser o mais correto, numa situação de contexto profissional o dia é focado na implementação desta plataforma como projeto a tempo inteiro, e em constante comunicação com os stakeholders, algumas estatísticas não foram possíveis de realizarem-se, nomeadamente das métricas por turmas e do aluno em específico. Também não foi possível fazer testes de stress nem verificar a capacidade de resposta do servidor em situações críticas.

Para trabalhos futuros, deverá ser analisada a hipótese de substituir a versão do arduíno mencionado para a versão mais recente, com placa WiFi embutida, descartando assim, a necessidade de um minicomputador a fazer de controlador para obter uma redução nos custos associados ao projeto.

Será recomendado também planear e implementar a integração deste serviço com os restantes sistemas da ULHT, nomeadamente a secretaria virtual (NetPA), para obter um horário completo das aulas do professor, dados adicionais dos alunos que possam se mostrar relevantes para a análise, a pesquisa dos mesmos e realizar o processo de marcação de faltas na unidade curricular automaticamente. Também se pode averiguar o uso da aplicação mobile “Ensino Lusófona” com leitura de QRCode gerado para a aula e assim marcar a presença do aluno diretamente sem depender de associações e dependência de dispositivos com RFID.

● Bibliografia

- [DEISI21] DEISI, Regulamento de Trabalho Final de Curso, Set. 2021.
- [TaWe20] Tanenbaum, A. e Wetherall, D., *Computer Networks*, 6ª Edição, Prentice Hall, 2020.
- [ULHT21] Universidade Lusófona de Humanidades e Tecnologia, www.ulusofona.pt, acessado em Out. 2021.
- [Muffin, 2016] S. M. Čisar, R. Pinter, V. Vojnić, V. Tumbas and P. Čisar, "Smartphone application for tracking students' class attendance," 2016 IEEE 14th International Symposium on Intelligent Systems and Informatics (SISY), 2016, pp. 227-232, doi: 10.1109/SISY.2016.7601502.
- [HFR, 2016] S. Lukas, A. R. Mitra, R. I. Desanti and D. Krisnadi, "Student attendance system in classroom using face recognition technique," 2016 International Conference on Information and Communication Technology Convergence (ICTC), 2016, pp. 1032-1035, doi: 10.1109/ICTC.2016.7763360.
- [SiGeFA, 2019] Evariste, M., & Sousa, B. (n.d.). *SIGEFA - Sistema de Gestão de frequências de alunos - universidade aberta*. Retrieved November 23, 2022, from <https://journals.uab.pt/index.php/rcc/article/download/181/145/>
- [PYTHON] Python 3.10 Release, <https://www.python.org/downloads/release/python-3100/>
- [FLASK] Python Flask, <https://flask.palletsprojects.com/en/2.2.x/changes/#version-2-2-2>
- [SQLITE] SQLite 3.40.0, <https://www.sqlite.org/index.html>
- [ARDUINO] Arduino, <https://www.arduino.cc>
- [JWT] JSON Web Tokens, <https://jwt.io/>
- [JSON] JSON, <https://www.json.org/json-en.html>
- [REACT] REACT Dev, <https://react.dev/>
- [RFID] RFID, https://pt.wikipedia.org/wiki/Identifica%C3%A7%C3%A3o_por_radiofrequ%C3%A4ncia
- [MFRC522] Módulo MFRC522, <https://www.arduino.cc/reference/en/libraries/mfrc522/>
- [SQLALCHEMY] SQLAlchemy, <https://www.sqlalchemy.org/>
- [SQL] SQL, <https://pt.wikipedia.org/wiki/SQL>
- [TFC] Unidade curricular "Trabalho Final de Curso" de LEI da ULHT, <https://www.ulusofona.pt/lisboa/licenciaturas/engenharia-informatica/ULHT260-5140>
- [MAUSER] Loja de eletrónica Mauser, <https://mauser.pt/>

● Anexo 1 – Questionário de viabilidade

1. Como marca as presenças na sala de aula?

● Faço a chamada verbalmente	4
● Peço para escreverem o nome e número numa folha	3
● Anoto apenas a quantidade de alunos na sala	5
● Não marco presenças	4
● Outro	4



2. Utiliza alguma ferramenta para controlar a assiduidade dos alunos e anotar as presenças das aulas? Se sim, qual?

8 inquiridos (47%) responderam **Excel** a esta pergunta.



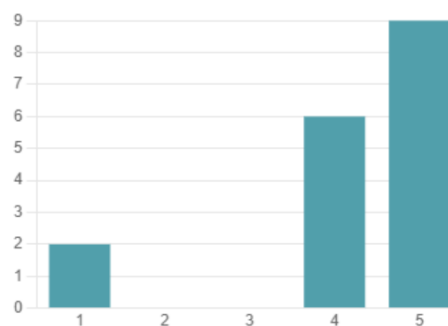
3. Quanto tempo de aula gasta, em média, para fazer a chamada/marcação de presenças?

● 0-5min	11
● 5-10min	4
● 10-15min	2
● 15-30min	0
● Mais de 30min	0



4. Como considera a importância de uma plataforma que otimizasse a marcação de presenças na sala de aula e listasse as mesmas de uma forma rápida e prática, para além de fornecer informação visual das faltas?

4.18
Classificação Média



● Glossário

LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
TFC	Trabalho Final de Curso
FUC	Ficha da Unidade Curricular
DEISI	Departamento de Engenharia Informática e Sistemas de Informação
ULHT	Universidade Lusófona de Humanidades e Tecnologias
TFC	Trabalho Final de Curso
RGPD	Regulamento Geral de Proteção de Dados
JSON	JavaScript Object Notation
JWT	JSON Web Tokens
REST	Representational State Transfer
API	Application Programming Interface
SQL	Structured Query Language
ORM	Object-Relational Mapping
RFID	Radio Frequency Identification
NFC	Near Field Communication
USB	Universal Serial Bus