



UNIVERSIDADE
LUSÓFONA

ULHT

Control Panel

DEISI278

Aluno: João Miguel Isidro Antas – a22002629

Aluno: Tiago Alexandre Pinto da Águeda – a22001757

Orientadores: Prof. Dr. João Pedro Leal Abalada De Matos Carvalho e

Prof. Dr. João Pedro Calado Barradas Branco Pavia

Trabalho Final de Curso | Engenharia Informática | 30/06/2023

Direitos de cópia

ULHT Control Panel, Copyright de João Miguel Isidro Antas e Tiago Alexandre Pinto da Águeda, 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 planeamento do ano letivo é uma época de grande esforço e trabalho por parte de todos os seus intervenientes, principalmente ao nível da organização dos horários académicos. A grande variedade de constrangimentos associados a diferentes necessidades no seio da comunidade académica tem implicações na eficiência da utilização das ferramentas de auxílio que permitem a realização desta tarefa.

No âmbito da Unidade Curricular de Trabalho Final de Curso (TFC), é solicitado aos alunos a criação de algo útil e inovador para o meio empresarial ou para o meio universitário geral. Intitulado de “ULHT Control Panel”, este projeto aborda o problema da criação de uma agenda académica, com foco no Departamento de Engenharia Informática e Sistemas de Informação (DEISI) e posteriormente nas diversas escolas e faculdades que a Universidade Lusófona incorpora. Este trabalho incide na avaliação desta abordagem e no desenvolvimento de uma ferramenta capaz de organizar toda a agenda académica, melhorando e utilizando todas as capacidades das ferramentas de trabalho do Excel, implementadas no modo atual e, para além disso, ultrapassando-as.

Este projeto consiste na implementação de uma aplicação web tendo como objetivo primordial auxiliar o secretariado e os alunos durante todo o ano letivo, quer na formulação dos horários, quer requisição de salas.

Palavras-chave: horários académicos, ULHT Control Panel, organização, agenda académica, aplicação web

Abstract

The planning of the academic year is a time of great effort and work by all involved, especially in terms of the organization of academic schedules. The wide variety of constraints associated with different needs within the academic community has implications on the efficiency of the aid tools that enable this task to be carried out.

Within the scope of the Final Coursework Curricular Unit (TFC), students are asked to create something useful and innovative for the business world or for the general university environment. Entitled "ULHT Control Panel", this project addresses the problem of creating an academic agenda, focusing on the Department of Computer Engineering and Information Systems (DEISI) and, subsequently, on the various schools that the Universidade Lusófona incorporates. This work focuses on the evaluation of this approach and the development of a tool capable of organizing the entire academic agenda, improving and using all the capabilities of the Excel working tools, implemented in the current mode and, furthermore, surpassing them.

This project consists of the implementation of a web application with the primary objective of assisting the secretariat and students throughout the academic year, both in the formulation of schedules and the requisition of rooms.

Keywords: academic schedules, ULHT Control Panel, organization, academic agenda, web application

Índice

Resumo	iii
Abstract.....	iv
Lista de Figuras	7
Lista de Tabelas	9
1 Introdução.....	10
2 Identificação do Problema.....	11
3 Viabilidade e Pertinência	13
4 Benchmarking	19
5 Engenharia.....	23
5.1 Levantamento e Análise dos Requisitos	23
5.1.1 Requisitos Funcionais:.....	24
5.1.2 Requisitos Não Funcionais:	27
5.1.3 Requisitos Técnicos:	28
6 Estrutura da aplicação	29
6.1 Diagrama de casos de uso	29
6.2 Diagrama Entidade-Relação	30
6.3 Estrutura da aplicação	32
7 Solução Proposta.....	33
7.1 Arquitetura do Sistema	33
7.2 Back-end	34
7.3 Front-end.....	35
7.4 Melhorias de Usabilidade	35
7.5 Abrangência	36

7.6	Código Fonte e Demonstração	37
8	Método e Planeamento	38
9	Plano de testes e validação	41
10	Resultados.....	43
10.1	Update: 27 de janeiro de 2023	43
10.2	Update: 30 de março de 2023	46
10.3	Update: 25 de junho de 2023	48
11	Conclusão	53
11.1	Trabalhos Futuros	53
12	Bibliografia.....	54
13	Anexos	55
13.1	Formulários	55
13.2	Progresso de trabalho.....	55
13.3	Documentação dos serviços	56
14	Glossário	57

Lista de Figuras

Figura 1 - Construção, distribuição horários e calendarização de exames de uma Instituição de Ensino Superior	11
Figura 2 - Resultados relativos à satisfação do método vigente pela coordenação	14
Figura 3 - Resultados do formulário relativamente à sobreposição de dois turnos	14
Figura 4 - Resultados do formulário relativamente à sobreposição de salas	14
Figura 5 - Resultados relativos à satisfação do método vigente pelos docentes	15
Figura 6 - Resultados relativos ao método de marcação e tempo de espera de salas	15
Figura 7 - Resultados relativos à opinião dos docentes sobreposição e marcações de salas	16
Figura 8 - Resultados relativos à opinião dos docentes sobre a benefícios da aplicação	16
Figura 9 - Resultados relativos à satisfação do método vigente pelos alunos	17
Figura 10 - Resultados relativos à marcação e opinião no agendamento de salas	17
Figura 11 - Resultados relativos ao interesse dos alunos em utilizar a aplicação	17
Figura 12 - Resultados relativos à sobreposição na marcação de salas	18
Figura 13 - Plataforma GOP – IST	20
Figura 14 - Formatos de Impressão exemplo no programa de horário gp-Untis	21
Figura 15 - Diagrama de Use Case	29
Figura 16 - Diagrama E.R.	31
Figura 17 – Diagrama em árvore	32
Figura 18 - Arquitetura do Sistema	33
Figura 19 - Funcionalidades dos agendamentos no calendário	36
Figura 20 - Cronograma proposto em formato Gantt	38
Figura 21 - Cronograma proposto em formato Gantt detalhado	39
Figura 22 - Login e Recupera Senha	43
Figura 23 - Visualização do calendário (Mês e Semana)	44
Figura 24 - Visualização do calendário (Dia e Agenda)	45
Figura 25 - Página de requisição de salas	46
Figura 26 - Visualização do calendário	47
Figura 27 - Pop-up de Criação, Edição e Eliminação	47
Figura 28 - Página de administração	48
Figura 29 – Página do agendamento	49

Figura 30 – Página de requisição de salas	49
Figura 31 – Página de notificações (elementos do secretariado).....	50
Figura 32 – Página de notificações (aprovação de requisição)	50
Figura 33 – Página de notificações (utilizador).....	51
Figura 34 – Página da administração.....	51
Figura 35 – Página da administração (Editar)	52
Figura 36 - Progresso de trabalho.....	55
Figura 37 - Exemplo da documentação do serviço (POST)	56

Lista de Tabelas

Tabela 1 - Exemplo de Tabela atual de criação de horário	19
Tabela 2 - Análise de Benchmarking	22
Tabela 3 - Requisitos Funcionais.....	27
Tabela 4 - Requisitos Não Funcionais	28
Tabela 5 - Requisitos Técnicos.....	28
Tabela 6 - Tabela de testes	42

1 Introdução

Todos os estabelecimentos de ensino no início de cada ano letivo deparam-se com a necessidade da criação de horários académicos.

A criação de um horário académico é uma tarefa de difícil concretização visto que o secretariado se vê obrigado a executar diversas etapas para a criação dos respetivos horários. Tratando-se de algo que tem como objetivo fazer uma gestão eficaz e eficiente dos horários académicos, bem como das marcações de salas de aula, esta aplicação concentra em si todas estas funcionalidades, tendo em consideração diversas restrições impostas pelo secretariado, docentes e alunos.

A conjugação de sistemas capazes de gerir informações académicas, combinados com técnicas eficientes, podem potenciar a criação de horários académicos de forma semiautomática, como o desempenho que na maioria dos casos não pode ser alcançado através das técnicas tradicionais.

2 Identificação do Problema

A era digital surgiu com o intuito de facilitar e automatizar os processos digitais e tecnológicos ao nível de todas as vertentes que todos nós conhecemos, desde o mundo empresarial à saúde, ensino, entre outros. A implementação e uso da tecnologia é transversal ao mundo digital no qual todos nós, cidadãos do mundo, nos inserimos, sendo também uma ferramenta de combate a diversos problemas que surgem.

Deste modo, iniciámos, de forma proativa, um levantamento de problemas ou possíveis oportunidades de melhoramento no meio universitário onde nos inserimos. Assim, identificou-se uma excelente oportunidade de melhoria no que diz respeito à organização e planeamento dos horários académicos. Tal como é do conhecimento geral, a conceção e atribuição de turmas, a requisição de salas, o agendamento de exames e/ou testes é ainda efetuada de uma forma pouco prática e que, na grande maioria das vezes, conduz a alguns percalços. Através do esquema apresentado [Figura 1], podemos de facto, visualizar o modo como o processo descrito se procede atualmente.

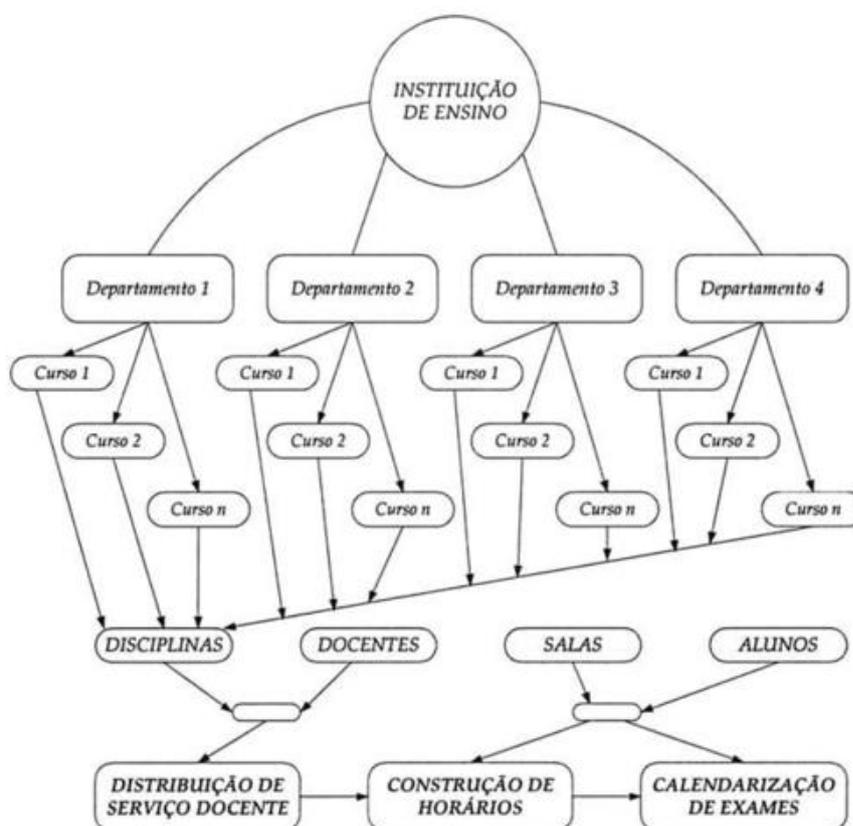


Figura 1 - Construção, distribuição horários e calendarização de exames de uma Instituição de Ensino Superior

Neste sentido, e por forma a colmatar todos os potenciais problemas supracitados anteriormente, decidimos desenvolver uma aplicação web com o intuito de auxiliar os docentes na organização e planeamento do calendário escolar. Atualmente esta tarefa é concretizada através da utilização de ferramentas de trabalho como o Excel, o que conduz a uma gestão ineficiente no âmbito da alteração de horários, podendo afetar também a marcação de salas e a conciliação entre docentes.

A implementação desta nova funcionalidade na Universidade Lusófona contribuirá para a resolução de problemas que os docentes se veem confrontados, no planeamento de um novo ano letivo, bem como auxiliar também os alunos na requisição de salas para efeitos de estudo ou elaboração de trabalhos.

3 Viabilidade e Pertinência

Através da elaboração de formulários anónimos (com auxílio da plataforma Google Forms) a docentes, elementos da coordenação do Departamento de Engenharia Informática e Sistemas de Informação (DEISI) e a alunos, conseguimos obter respostas às nossas questões relativamente à atribuição de turmas, a metodologia de requisição de salas, o agendamento de exames e/ou testes e a metodologia utilizada na gestão da atividade do departamento.

Data de início: 25/10/2022

Data de *terminus*: 15/11/2022

Número de resposta obtidas de docentes: 23

Número de resposta obtidas de elementos de coordenação do departamento: 5

Número de resposta obtidas de alunos: 34

Após analisarmos as respostas obtidas nos formulários fornecidos a 5 pessoas que fazem parte da coordenação do departamento, podemos concluir que:

- O método vigente pode ser melhorado [Figura 2];
- 4 desses elementos confirmam que já aconteceu uma sobreposição entre dois turnos [Figura 3];
- 2 desses elementos disseram que a resolução do problema mencionado anteriormente demorou 1 dia e 2 outros elementos relatam ter demorado até 2 dias a ser resolvido [Figura 4].

Como se sente em relação ao método de marcação vigente?

5 respostas

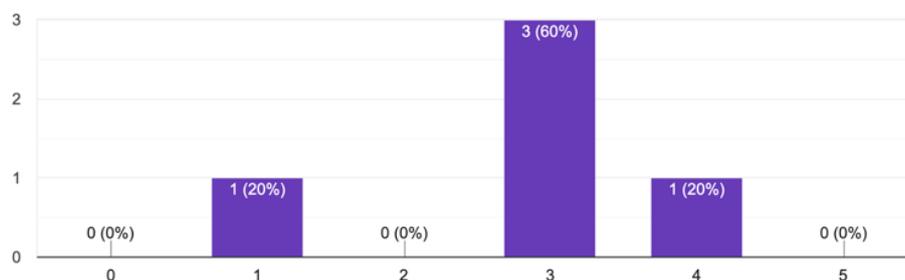


Figura 2 - Resultados relativos à satisfação do método vigente pela coordenação

Já aconteceu, por uma questão de falha humana ou logística, haver uma sobreposição de dois turnos de alunos à mesma hora?

5 respostas

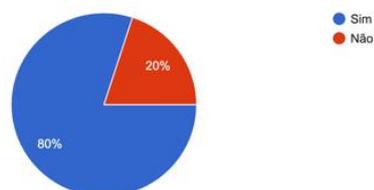


Figura 3 - Resultados do formulário relativamente à sobreposição de dois turnos

Quanto tempo em média demora a resolução de um problema como a sobreposição de salas?

5 respostas

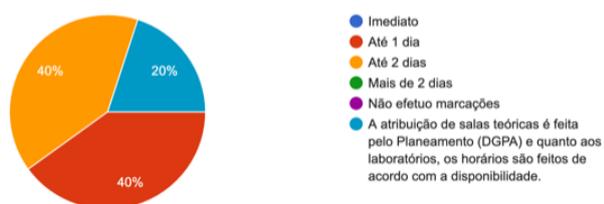


Figura 4 - Resultados do formulário relativamente à sobreposição de salas

Analisámos de igual modo as respostas obtidas nos formulários fornecidos a vários de docentes do DEISI, podendo concluir que:

- Cerca de 50% dos inquiridos não se sente muito satisfeito em relação ao método de marcação vigente [Figura 5];

- O processo de marcação de salas por parte dos docentes é realizado de diferentes modos (e-mail, chamada telefónica, através do PUI e online) [Figura 6];

- Cerca de 50% dos docentes questionados deparam-se com um tempo de espera de 1 e 2 dias para efetuar marcações e 9% com tempos de espera superiores a estes [Figura 6];

- Mais de 70% dos docentes que foram inquiridos admite já ter sido confrontado com uma marcação de sala com outro docente [Figura 7] assim como também já se ter deparado com a sobreposição de dois turnos de alunos ao mesmo tempo [Figura 7];

- Mais de 80% acredita que uma aplicação web iria facilitar este processo no sentido de evitar a ocorrência deste tipo de problemas, sendo que 13,6% não acredita no seu benefício [Figura 8].

Como se sente em relação ao método de marcação vigente?

23 respostas

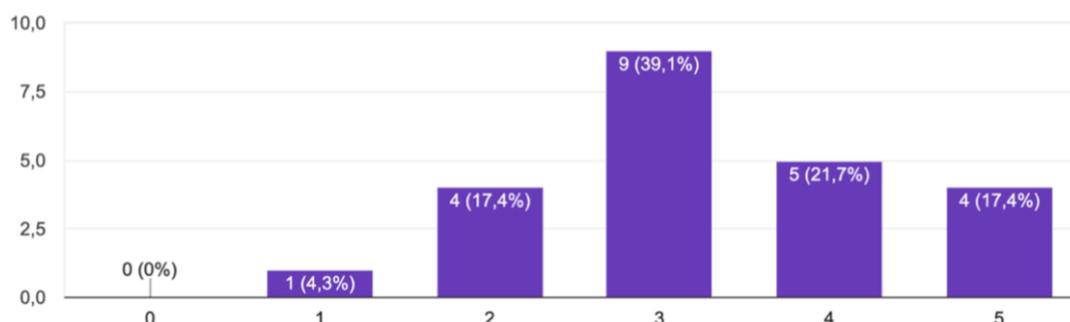
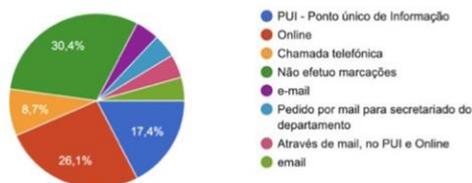


Figura 5 - Resultados relativos à satisfação do método vigente pelos docentes

Como é efetuada a marcação de salas?

23 respostas



Quanto tempo em média demora o pedido de marcação de sala a ser validado?

23 respostas

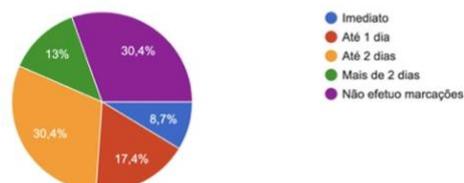


Figura 6 - Resultados relativos ao método de marcação e tempo de espera de salas

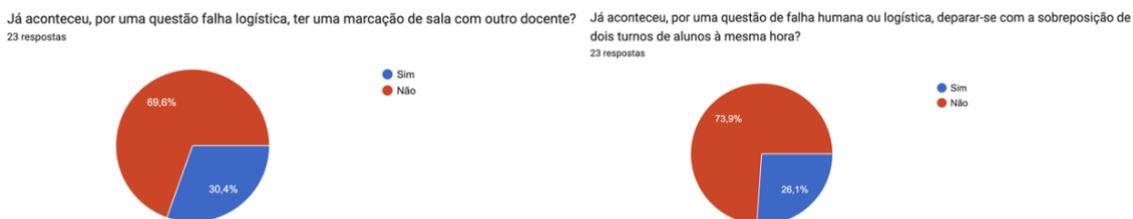


Figura 7 - Resultados relativos à opinião dos docentes sobreposição e marcações de salas

Na sua opinião, a implementação de uma aplicação web para facilitar/auxiliar a marcação dos horários e na gestão dos mesmos, facilitaria a resolução destes ou outros problemas?
23 respostas

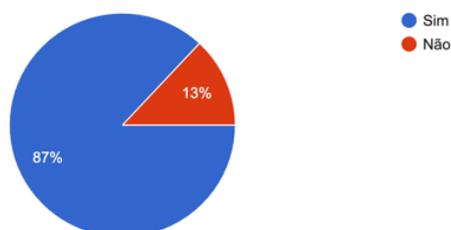


Figura 8 - Resultados relativos à opinião dos docentes sobre a benefícios da aplicação

Por fim, analisando as respostas obtidas nos formulários fornecidos a alguns alunos do DEISI, podemos concluir que:

- mais de 50% dos inquiridos não se sente muito satisfeito em relação ao método de marcação implementado atualmente [Figura 9];

- mais de 75% dos alunos questionado não sabe que pode fazer marcações de salas para estudo ou outras finalidades académicas [Figura 10];

- Cerca 90% acredita que a implementação de uma aplicação web seria uma boa solução para o agendamento de salas para fins de estudo [Figura 10];

- Cerca de 75% teria muito interesse em utilizar uma aplicação web para marcações de salas [Figura 11];

- Cerca de 23% já se deparou com uma sala ocupada mesmo tendo efetuado a sua marcação previamente [Figura 12].

Como se sente em relação ao método de marcação implementado atualmente?
34 respostas

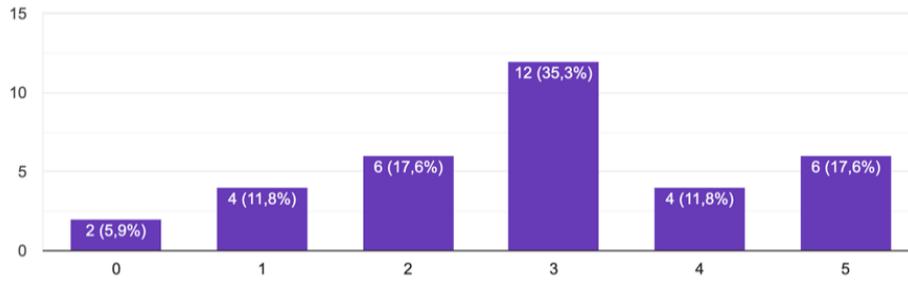
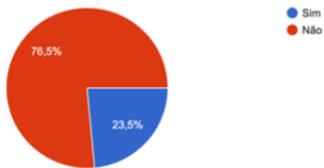


Figura 9 - Resultados relativos à satisfação do método vigente pelos alunos

Sabias que podes fazer a marcação de salas para estudo ou outra finalidades académicas?
34 respostas



Achas que uma aplicação web seria uma boa solução para o agendamento de salas para fins de estudo?
34 respostas

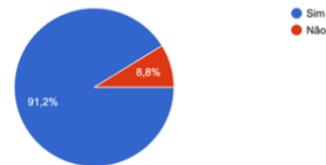


Figura 10 - Resultados relativos à marcação e opinião no agendamento de salas

Qual seria o teu interesse em utilizar uma aplicação web para a marcação de salas?
34 respostas

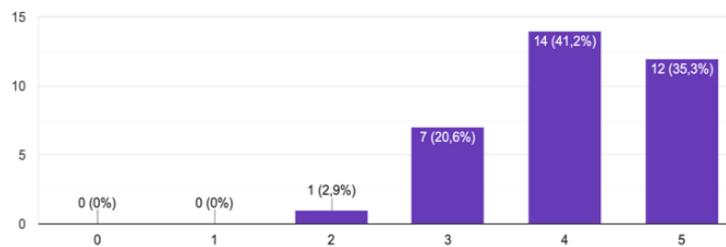


Figura 11 - Resultados relativos ao interesse dos alunos em utilizar a aplicação

Já aconteceu, por questão de falha logística, deparares-te com uma sala ocupada, tendo efetuado a sua marcação previamente?
34 respostas

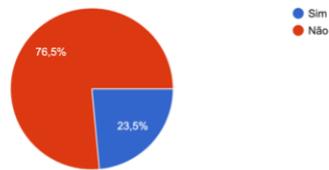


Figura 12 - Resultados relativos à sobreposição na marcação de salas

A construção destes formulários [anexados na secção 11.1 *Anexos*] foi importante não só para percebermos se de facto as nossas opiniões em relação ao problema exposto eram comuns, como também motivou para a realização desta aplicação e perceber melhor o que era pretendido e esperado pela comunidade académica.

4 Benchmarking

A aplicação ULHT Control Panel, distingue-se das restantes que a Universidade Lusófona possui, pois tem a funcionalidade de visualizar e criar os horários académicos, proceder à reserva de salas de aulas, seleção de horários para cada docente e criação das turmas, tudo numa única aplicação, gerando automaticamente os horários em formato CSV (Comma-separated values) sem recorrer a outras funcionalidades que necessitam de muito tempo na sua elaboração [Tabela 1].

Curso: Engenharia Informática						Ano Lectivo:2022-2023	
Plano: 107						Semestre 1º	
Local: ULHT						Ano 3º	
						Turma	
	segunda	terça	quarta	quinta	sexta		
08.00 - 08.30	Data Science LD01EINF01	IHM LD01EINF02	Computação Distribuída TD01EINF01		Comp Distribuída PD01EINF01	Engº Software LD01EINF03	
08.30 - 09.00	Ricardo Sousa F,3,3	Miguel Tavares F,2,6	Paulo Guedes SALA		Sergio Nunes F,3,3	Rui Santos F,2,6	
09.00 - 09.30							
09.30 - 10.00							
10.00 - 10.30	IHM LD01EINF01	Data Science LD01EINF02	Data Science TD01EINF01		Interação Humano-Máquina TD01EINF01	Engº Software LD01EINF01	
10.30 - 11.00	Miguel Tavares F,2,6	Ricardo Sousa F,3,3	M.Marques Pita SALA		Conceição Costa SALA	Pedro Perdigão F,2,6	
11.00 - 11.30							
11.30 - 12.00							
12.00 - 12.30	Data Science LD01EINF03				Engenharia de Software TD01EINF01	Engº Software LD01EINF02	
12.30 - 13.00	Ricardo Sousa F,3,3				José Cascais Brás SALA	Rui Santos F,2,6	
13.00 - 13.30						Comp Distribuída PD01EINF03	
13.30 - 14.00						Sergio Nunes Q,3,1	
14.00 - 14.30							
14.30 - 15.00							
15.00 - 15.30							
15.30 - 16.00							
16.00 - 16.30							
16.30 - 17.00							
17.00 - 17.30							
17.30 - 18.00							
18.00 - 18.30							
18.30 - 19.00	Data Science LN01EINF01	Data Science TN01EINF01	TFC - Seminários Turmas diurnas e noturnas Armando Vieira S.O.10	Interação Humano-Máquina TN01EINF01	IHM LN01EINF01		
19.00 - 19.30	Ricardo Sousa F,3,3	M.Marques Pita SALA		Conceição Costa SALA	Miguel Tavares F,2,6		
19.30 - 20.00							
20.00 - 20.30	Comp Distribuída PN01EINF01	Engº Software LN01EINF01		Computação Distribuída TN01EINF01			
20.30 - 21.00	Sergio Nunes F,3,3	Pedro Perdigão F,2,6		Paulo Guedes SALA			
21.00 - 21.30							
21.30 - 22.00							
22.00 - 22.30							
22.30 - 23.00							
23.00 - 23.30							
23.30 - 24.00							

Tabela 1 - Exemplo de Tabela atual de criação de horário

Analisando as soluções existentes no mercado no âmbito da construção de horários, constata-se a existência de diversos tipos de software, como o gp-Untis, capazes de apresentar um processo automático de construção de horários. A apresentação gráfica das aplicações é variada, dependendo por vezes da metodologia que foi adotada na construção dos horários.

De realçar também a criação de gabinetes para a resolução de situações deste género, como o Gabinete de Organização Pedagógica (GOP) [Figura 13] do Instituto Superior Técnico da Universidade de Lisboa, que tem como objetivo a resolução de assuntos como:

- Gerir a utilização e proceder à reserva de salas para aulas, provas de avaliação ou outras atividades;
- Supervisionar a resolução de problemas relacionados com a sobrelotação de turmas durante o processo de inscrição dos alunos;
- Gerir os meios audiovisuais de apoio às atividades de ensino ou outras.

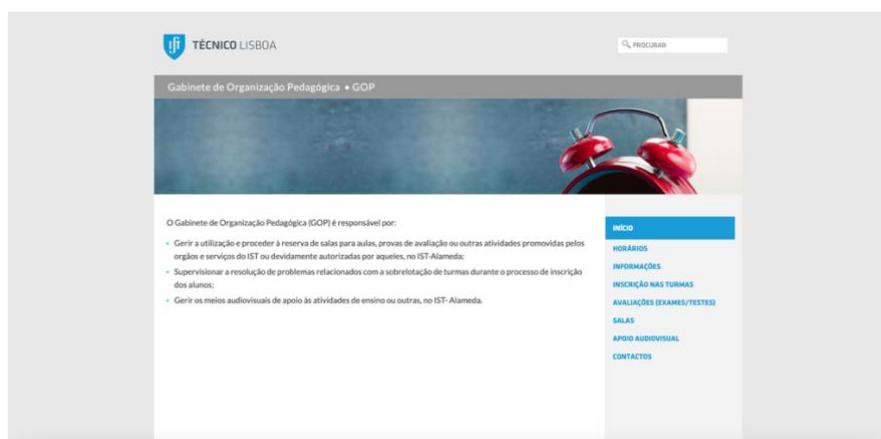


Figura 13 - Plataforma GOP – IST

O GOP ajuda na resolução de muitos problemas, contudo apenas recorrendo ao uso excessivo de recursos humanos.

O gp-Untis trata-se de uma aplicação desenvolvida por uma empresa de análise informática, onde o seu foco está nas escolas básicas e secundárias portuguesas [Figura 14]. O software utilizado tem uma apresentação em ambiente Windows, onde é necessário alguma habituação e experiência por parte do utilizador devido à complexidade do ambiente apresentado.

Algumas das suas funcionalidades mais utilizadas são:

- Distribuição de serviço docente;
- Implementação de uma gestão curricular flexível;
- Diferentes grelhas semanais;
- Envio por mail dos horários aos docentes.

	Segunda	Terça	Quarta	Quinta	Sexta
8.20 9.05					
9.05 9.50					
10.05 10.50				AP 5ºF 149	
10.50 11.35					
11.50 12.35		PPCA6			
12.35 13.20					
13.35 14.20	RC 5ºF 214	PLIAP 6ºF 201	ING 5ºF 214	DT DT	
14.20 15.05	ING 6ºF 219	CDT		ING 6ºPCA 230	
15.15 16.00	ING 5ºF 214	LP 5ºF 214	CDT	ING 6ºF 201	
16.00 16.45			ING 6ºPCA 224		
17.00 17.45	LP 5ºF 214	DT DT	CDT	Apoio Apob 221	
17.45 18.30					

TL	Disciplina	Turma/s	Texto	C. Não Lectiva
4	Língua Portuguesa	5ºF		
2	Inglês	6ºPCA		
3	Inglês	6ºF		
4	Inglês	5ºF		
2	Área de Projecto	5ºF		
1	Formação Cívica	5ºF		
16.00				0
1	Plano de Leitura/Apoio	6ºF	Horas Escola	CIL
4	C. Directores de Turma		Artigo 79	CIL
2	Director de Turma	DT	DT: 5ºF	CIL
1	Reuniões PCA6		Horas Escola	CIL
1	Apoio Pedagógico	Apoio	Supervenientes	CIL
9.00				5
2			Reuniões	TIR
9			Trabalho individual	TIR
11.00				2

Figura 14 - Formatos de Impressão exemplo no programa de horário gp-Untis

Com a evolução do mercado é exigido que cada vez mais os sistemas incluam para além da construção de horários, também os sistemas de distribuição de serviço docente, calendarização de exames ou agendamentos de salas. São requeridos também, sistemas que contenham processos integrados de construção semiautomática e automática de horários (por exemplo, a pessoa responsável pela construção dos horários de uma instituição, pode pretender um sistema que lhe permita construir de forma semiautomática os horários dos docentes e dos alunos, e após concluída essa fase, o sistema encarregar-se-ia de atribuir as salas a todas as aulas).

Na tabela seguinte [Tabela 2], será possível analisar de uma forma mais sistemática quais as diferentes funcionalidades dos softwares implementadas no mercado, em comparação com a aplicação web em desenvolvimento.



ULHT CONTROL PANEL

FEATURES	REQUISITAR SALAS	VISUALIZAR HORÁRIO	CRIAR HORÁRIO	NOTIFICAÇÕES PERSONALIZADAS
Metodo Atual		✓		
GOP	✓	✓	✓	
GP-Untis		✓	✓	
ULHT Control Panel	✓	✓	✓	✓

Tabela 2 - Análise de Benchmarking

5 Engenharia

Atualmente a elaboração de uma aplicação web, assim como de qualquer projeto no desenvolvimento empresarial necessita que seja feita uma ligação entre a equipa de desenvolvimento e a organização. Posto isto, foi implementado a criação de uma correta e precisa definição de requisitos de sistema, com o objetivo de guiar todos os elementos da equipa de trabalho como os gestores de projeto e desenvolvedores web durante o tempo definido para o projeto. Com a finalidade de a universidade se tornar mais eficiente na organização de horários, para que deste modo possam os docentes e alunos se sentirem mais motivados e organizados para o começo do novo ano letivo na sua universidade, bem como ambos conseguirem conciliar o equilíbrio entre a vida profissional e pessoal.

5.1 Levantamento e Análise dos Requisitos

Assim, como fase inicial deste projeto, resolvemos reunir um conjunto de requisitos [Tabela 3, 4, 5] que serão implementados para responder às necessidades dos respetivos utilizadores, fazendo cumprir o seu objetivo.

Analisando e discutindo sobre os respetivos requisitos que haveriam de ser implementados e respeitados, formaram-se os seguintes:

5.1.1 Requisitos Funcionais:

Requisito	Escala de tamanhos	Escala 'Have'	MD
RF1. Como utilizador, quero poder aceder à aplicação através do <i>login</i> onde terei de preencher o <i>e-mail</i> e <i>password</i> pessoal.	M	must have	3
RF2. Como utilizador, quero ter a possibilidade de recuperar a minha <i>password</i> através do <i>e-mail</i> .	S	nice to have	1
RF3. A aplicação não deverá permitir que os utilizadores não logados acedam a páginas privadas (Requisitar Sala, Horário, Agendamento, Administração, Notificação).	S	must have	1
RF4. Como utilizador, quero poder efetuar <i>logout</i> através de um botão no <i>dashboard</i> de modo a sair da minha conta.	S	must have	1
RF5. Como aluno ou docente, quero ter a possibilidade de efetuar a requisição de uma sala.	L	must have	4
RF6. Como utilizador, quero poder observar o ponto de situação das minhas requisições (“Em Análise”, “Aprovado” e “Recusado”).	S	nice to have	2
RF7. Como utilizador, quero ter a possibilidade de eliminar notificações recusadas pelo secretariado, através de um botão.	S	should have	2
RF8. Como elemento do secretariado, quero ter a possibilidade de aprovar ou recusar requisições feitas por alunos e docentes.	M	must have	3
RF9. Como elemento do secretariado, ao clicar no botão de aprovação quero ter a possibilidade de visualizar as salas de aulas livres de acordo com as necessidades da requisição (sala de computadores/ sem computadores).	M	nice to have	3
RF10. Como elemento do secretariado, ao clicar no botão para recusar a requisição do pedido, quero poder visualizar um <i>pop-up</i> de confirmação para evitar falsos cliques.	M	should have	3
RF11. Como elemento do secretariado e docente, pretendo filtrar o calendário através do curso, ano e/ou turma.	L	must have	5
RF12. Como utilizador, ao seleccionar os filtros deverei ter a possibilidade de limpar os mesmos clicando no respetivo botão.	S	nice to have	1

RF13. Como elemento do secretariado, quero poder gerar ficheiros em formato Excel com os horários de cada turma.	L	must have	4
RF14. Como elemento do secretariado, ao tentar criar uma aula sem uma turma selecionada deverá aparecer um <i>Toast</i> que me lembre que deverei selecionar uma turma.	S	nice to have	1
RF15. Como utilizador, quero ter a possibilidade de visualizar agendamentos por mês, semana ou dia.	M	nice to have	3
RF16. A aplicação deverá impossibilitar selecionar o filtro da turma caso o filtro do ano não esteja selecionado, assim como só selecionar o filtro do ano apenas quando o filtro do curso estiver selecionado.	S	should have	1
RF17. Como utilizador quero ter a possibilidade, após terminar de navegar no calendário, voltar ao dia corrente através de um botão.	S	nice to have	1
RF18. A aplicação deverá exibir no calendário cada unidade curricular com cores diferentes para uma melhor consulta e distinção.	M	nice to have	3
RF19. Como utilizador, ao clicar numa unidade curricular deverá aparecer a informação detalhada da mesma através de um <i>pop-up</i> .	L	must have	5
RF20. Como elemento do secretariado, ao clicar na unidade curricular deverei ter a possibilidade de editar ou eliminar o horário.	M	nice to have	2
RF21. Como elemento do secretariado, ao selecionar um horário deverá aparecer um <i>pop-up</i> com um formulário (unidade curricular, tipologia de aula, sala de aula, docente, data inicial, data final, hora inicial e hora final) que permita criar o horário.	L	must have	5
RF22. Como elemento do secretariado, ao criar um horário, apenas deverei visualizar as unidades curriculares da respetiva turma que ainda não tenham o respetivo horário.	M	nice to have	3
RF23. Como elemento do secretariado, no momento da criação do horário, ao selecionar a unidade curricular, apenas deverá aparecer a tipologia de aula que ainda não tenha sido atribuída a um horário da respetiva unidade curricular.	M	nice to have	3

RF24. Como elemento do secretariado, no momento da criação do horário, ao selecionar a unidade curricular e tipologia de aula, deverão aparecer apenas os docentes que tenham sido selecionados para lecionar a unidade curricular.	M	nice to have	3
RF25. Como elemento do secretariado, no momento da criação do horário, apenas deverão ser exibidas as salas de aula livres e que respeitem a tipologia de aula selecionada.	M	nice to have	3
RF26. Como elemento do secretariado, ao selecionar um horário livre deverá aparecer automaticamente no <i>pop.up</i> de criação, o horário inicial e final.	M	should have	3
RF27. Como elemento do secretariado, no momento da criação do horário e após o preenchimento do respetivo formulário, ao clicar no botão “Criar”, deverá aparecer um <i>Toast</i> a indicar se a criação foi ou não bem sucedida.	S	nice to have	1
RF28. Como elemento do secretariado, ao clicar no botão de edição de um horário, deverá aparecer um <i>pop-up</i> igual ao da criação de um horário, mas os campos deverão estar preenchidos com as informações do horário selecionado.	M	must have	3
RF29. Como elemento do secretariado, pretendo ter uma página que me possibilite a administração dos diversos dados do departamento (alunos, docentes, unidades curriculares, cursos, salas de aula).	L	should have	4
RF30. Como elemento do secretariado, quero ter a possibilidade de adicionar, eliminar e editar os dados dos alunos.	M	must have	3
RF31. Como elemento do secretariado, quero ter a possibilidade de adicionar, eliminar e editar os dados dos docentes.	M	must have	3
RF32. Como elemento do secretariado, quero ter a possibilidade de adicionar, eliminar e editar os dados das unidades curriculares.	M	must have	3
RF33. Como elemento do secretariado, quero ter a possibilidade de adicionar, eliminar e editar os dados dos cursos.	M	must have	3
RF34. Como elemento do secretariado, quero ter a possibilidade de adicionar, eliminar e editar os dados das salas de aula.	M	must have	3

RF35. Como elemento do secretariado, ao clicar no botão de criação em cada aba, deverá aparecer um <i>pop-up</i> com os <i>inputs</i> respetivos de acordo com o diagrama de Entidade-Relação.	L	must have	4
RF36. Como elemento do secretariado, após preencher o formulário de criação, ao clicar no botão “Criar”, deverá aparecer um aviso que indique se a criação foi ou não bem sucedida.	S	nice to have	1
RF37. Como elemento do secretariado, ao clicar no botão de edição em cada aba, deverá aparecer um <i>pop-up</i> igual ao de criação, preenchido com a respetiva informação.	M	must have	4
RF38. Como elemento do secretariado, após preencher o formulário de edição, ao clicar no botão “Editar”, deverá aparecer um aviso que indique se a edição foi ou não bem sucedida.	S	nice to have	4
RF39. Como elemento do secretariado, ao clicar no botão de eliminar em cada aba, deverá aparecer um <i>pop-up</i> de confirmação.	S	should have	1

Tabela 3 - Requisitos Funcionais

MD (*man-days*) referem-se à quantidade de esforço necessário para concluir uma tarefa ou projeto, normalmente medidos em termos do número de dias que uma pessoa levaria para concluir o trabalho, atribuído com base na complexidade e incerteza envolvidos na tarefa. Por exemplo, podemos atribuir 1 MD a uma tarefa de baixa complexidade e 5 MD a uma tarefa muito mais complexa.

Em termos práticos, este tipo de planeamento foi benéfico para o grupo gerir a sua organização e conseguir adquirir uma melhor perceção do que é necessário para a conclusão de cada etapa do projeto.

5.1.2 Requisitos Não Funcionais:

Requisito	Escala ‘Have’
RNF1. O utilizador deverá ter credenciais válidas, para conseguir entrar na sua conta pessoal.	must have

RNF2. O utilizador deverá conseguir consultar o horário para poder organizar a sua vida académica/profissional.	must have
RNF3. A aplicação deve respeitar a legislação em vigor, assim como as normativas da Comissão Nacional de Proteção de Dados (RGPD).	must have
RNF4. A aplicação deverá funcionar em qualquer <i>browser</i> .	must have
RNF5. No preenchimento de dados online a aplicação deve exigir aos utilizadores a aceitação da política de privacidade.	must have
RNF6. A aplicação deverá ter implementada uma cifra de encriptação nas passwords dos utilizadores da mesma.	must have

Tabela 4 - Requisitos Não Funcionais

5.1.3 Requisitos Técnicos:

Requisito	Escala 'Have'
RT1. A aplicação necessita de uma conexão à internet estável para poder ser acedida.	must have
RT2. A aplicação deverá conseguir aceder e recolher dados da base de dados.	must have
RT3. A aplicação deverá estar sempre atualizada.	must have

Tabela 5 - Requisitos Técnicos

6 Estrutura da aplicação

6.1 Diagrama de casos de uso

Através da implementação de um diagrama de casos de uso [Figura 15] consegue-se mostrar as interações esquemáticas entre o sistema e os seus ambientes. Sendo que, o referido diagrama, resume os detalhes dos usuários do seu sistema (também conhecidos como atores) e as interações deles com o sistema. Na formulação do respetivo diagrama foi usado um conjunto de símbolos e conectores especializados.

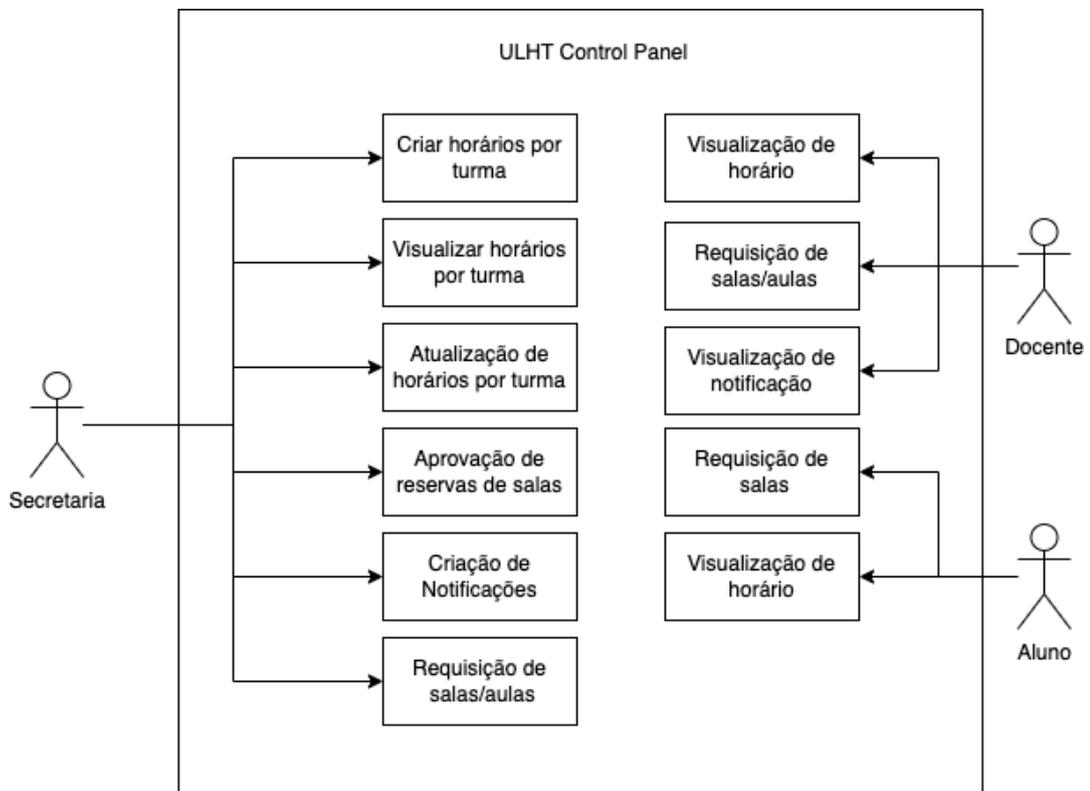
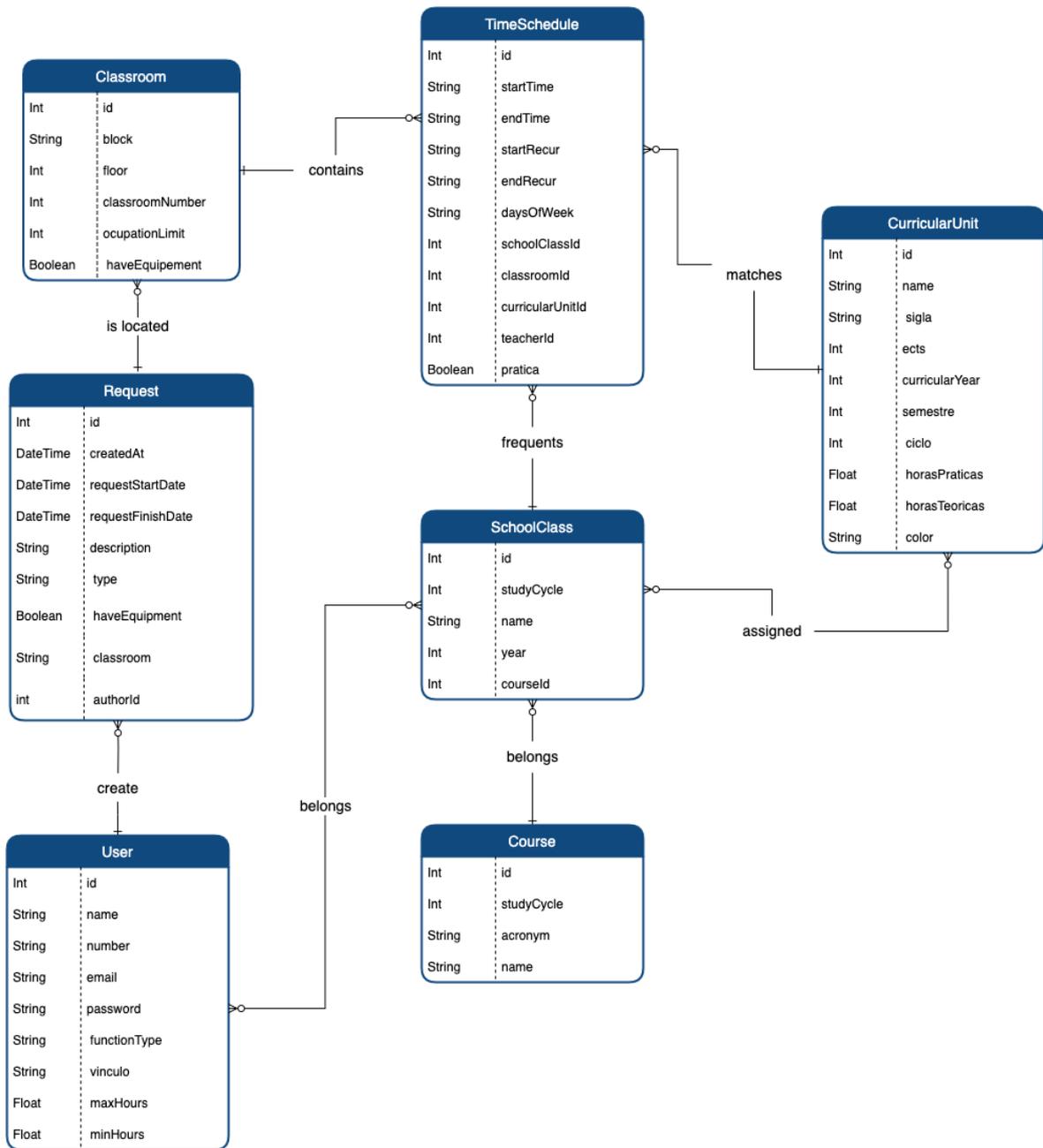


Figura 15 - Diagrama de Use Case

6.2 Diagrama Entidade-Relação

Os diagramas Entidade Relação (E.R) são uma forma de representação de entidades, relacionamentos e atributos. Estes diagramas são usados para modelar e criar bases de dados relacionais, em termos de regras lógicas e de negócio (num modelo lógico de dados) e em termos da tecnologia específica a ser implementada (num modelo físico de dados). Na engenharia de software, um diagrama E.R é muitas vezes um passo inicial na determinação de requisitos para um projeto de sistemas da informação, sendo também usado para modelar certas bases de dados. Uma base de dados relacional possui uma tabela relacional equivalente, e pode potencialmente ser expresso dessa forma, conforme necessário.

Deste modo, procedeu-se à elaboração do diagrama E.R [Figura 16], onde os retângulos representam as entidades e as linhas a conexão entre essas respectivas entidades.



§Figura 16 - Diagrama E.R

6.3 Estrutura da aplicação

Os diagramas em árvore [Figura 17] são úteis em projetos web porque ajudam a representar a estrutura de uma aplicação ou *website*. Podem ser usados para:

1. Representar a arquitetura de uma aplicação mostrando como diferentes componentes de uma aplicação web se relacionam entre si, incluindo a relação entre camadas de aplicação, como a camada de apresentação, de negócios e de dados;
2. Organizar a estrutura do *website*, podendo ser usados para mostrar a estrutura de pastas e arquivos de um *website*, incluindo a relação entre páginas e sub-páginas;
3. Identificar e resolver problemas de navegação podendo ajudar a identificar os mesmos, como *links* quebrados ou páginas que não são acessíveis;
4. Planear a escalabilidade, podendo ser utilizados para planear a escalabilidade de um *website*, identificando pontos críticos e possíveis soluções para problemas futuros.

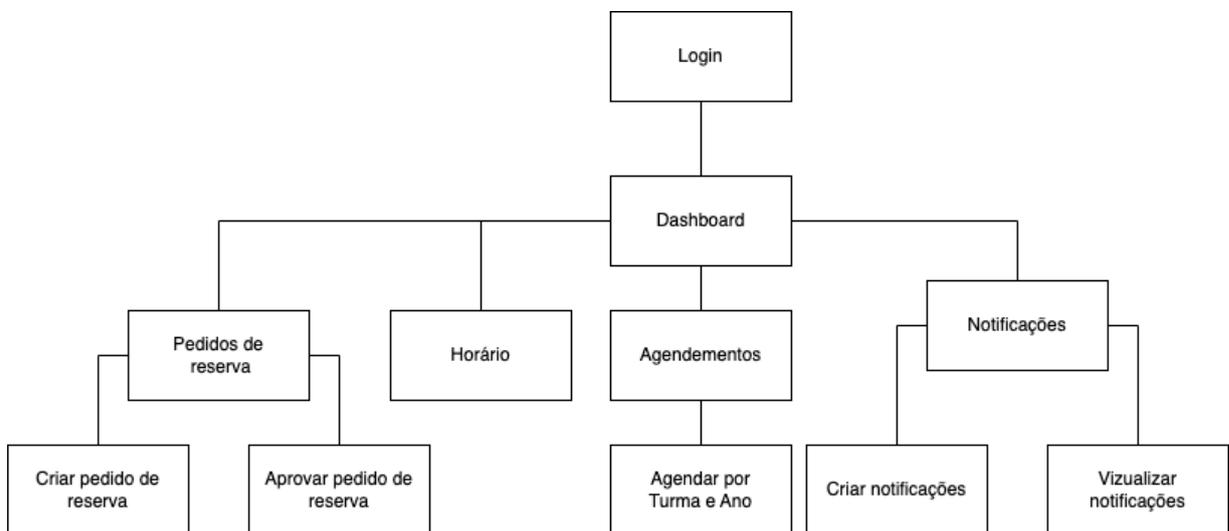


Figura 17 – Diagrama em árvore

7 Solução Proposta

7.1 Arquitetura do Sistema

A arquitetura, para além de ser um processo, é um produto que serve para orientar os gestores na criação de processos de negócio e os programadores de sistemas na criação de aplicações alinhadas com os objetivos e políticas de negócio. O processo de arquitetura compreende várias etapas que começam na ideia inicial e terminam num sistema verdadeiramente operacional.

Relativamente à arquitetura da aplicação web [Figura 18], o ULHT Control Panel contém uma componente *back-end* e uma componente *front-end*.

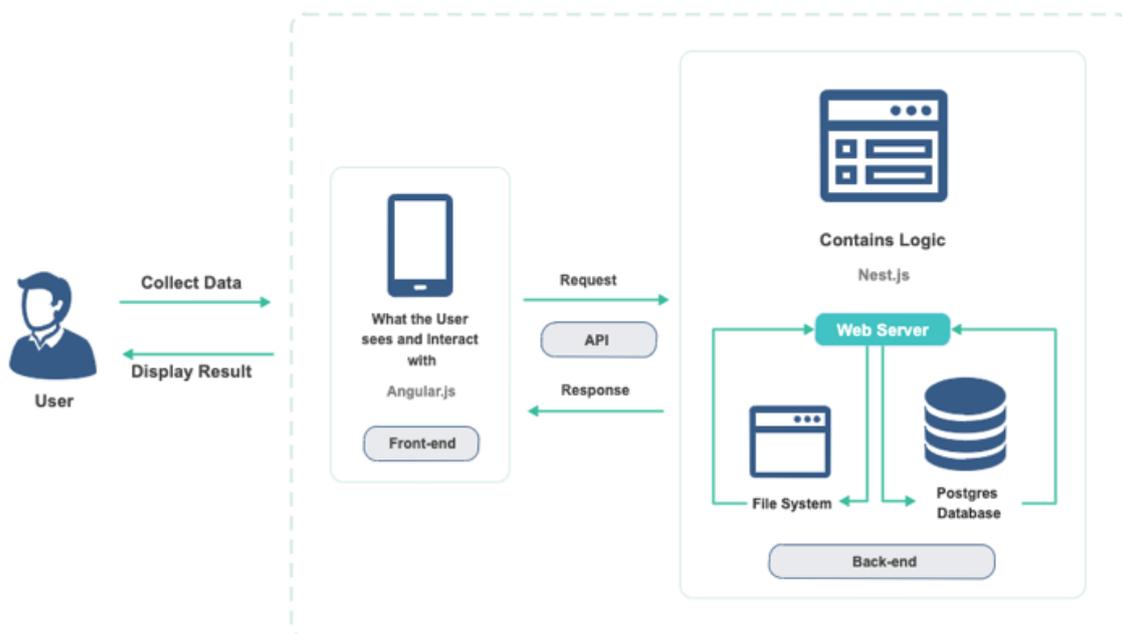


Figura 18 - Arquitetura do Sistema

Nesta arquitetura, o Nest.js é usado para construir a camada de serviço da aplicação, que expõe as APIs para o *front-end*. O Nest.js comunica com a base de dados através do Prisma, que gera automaticamente as consultas SQL e fornece uma API de consulta fácil de usar. O PostgreSQL é usado como a base de dados de *back-end*, armazenando todos os dados da aplicação. E o Angular.js é utilizado para construir a interface gráfica do utilizador, permitindo que os utilizadores interajam com a aplicação.

7.2 Back-end

Foi eleita a base de dados PostgreSQL com o intuito de aprendizagem de um novo sistema de gestão de base de dados, uma vez que que na unidade curricular de Base de Dados foi lecionado o MySQL. Foi decidido utilizar o Nest.js pois tem implementado o padrão CoC (*Convention Over Configuration*), ou seja, permite-nos estar mais preocupados com o modo como será desenvolvida a aplicação pois fornece por defeitos uma estrutura de pastas como forma de separar a lógica, do sistema. Utiliza TypeScript (língua programação desenvolvida pela Microsoft), o que nos permite trabalhar com tipagem e fazer a verificação de erros mais facilmente, o que o JavaScript não o permite. Possui uma arquitetura escalável, ou seja, se forem necessárias novas funcionalidades, o Nest.js torna mais fácil a implementação das mesmas.

Nest.js é um *framework* que permite a criação de APIs escaláveis e seguras usando JavaScript e TypeScript. É baseado em Express.js e oferece recursos avançados para lidar com rotas, *middlewares* e controladores. Isso permite que os desenvolvedores criem facilmente rotas para as suas APIs e gerenciem requisições e respostas. Além disso, o Nest.js tem suporte nativo para TypeScript, o que pode melhorar a qualidade do código e ajudar a encontrar erros de tempo de compilação.

Prisma é uma ferramenta de gestão de base de dados que permite definir um modelo de dados (*schema*) e gera automaticamente as consultas SQL para criar e gerir tabelas na base de dados. Fornece uma API de consulta fácil de utilizar para aceder aos dados na base de dados. Isso permite que os desenvolvedores trabalhem com os dados de forma mais intuitiva, sem precisar de escrever consultas SQL manualmente. Além disso, o Prisma fornece recursos avançados como migrações, transações e relacionamentos.

PostgreSQL é um sistema de gestão de base de dados relacional *open source*. É conhecido pela sua robustez, escalabilidade e segurança, e suporta uma variedade de tipos de dados, incluindo texto, números, datas e horas, entre outros.

7.3 Front-end

Desenvolvido através do *framework* Angular.js, da linguagem de marcação HTML e da linguagem de estilo CSS, que contém toda a estrutura e estilização da aplicação.

Angular.js é um *framework* JavaScript utilizado na construção da interface gráfica do utilizador, permitindo que os utilizadores interajam com a aplicação. Baseia-se em componentes para desenvolver a sua estrutura, e suporta o uso de JavaScript, HTML e CSS para a construção dos componentes. Isso permite que os desenvolvedores criem interfaces de utilizadores reutilizáveis, e também oferece recursos avançados como *data binding* e diretivas.

Os benefícios da utilização das tecnologias mencionadas anteriormente são apresentar uma documentação detalhada, o que permite a consulta, facilitando a aprendizagem, utilização e testagem. Contém uma arquitetura igual ao Nest.js, ou seja, é uma arquitetura baseada em componentes e utiliza o TypeScript.

7.4 Melhorias de Usabilidade

Como referido anteriormente o objetivo principal do ULHT Control Panel é melhorar o processo de criação de horários relativamente ao utilizado atualmente pelo secretariado. Após um *brainstorming* concluímos que para resolver o problema identificado seria importante termos como objetivo principal da aplicação a geração de ficheiros Excel que atualmente são criados manualmente, demorando mais tempo a executar esta tarefa e muito propício a erros.

Na página principal da aplicação [Figura 19] (Agendamentos) temos ao dispor um filtro que permite selecionar de uma forma fácil e eficaz a turma pretendida para a criação do horário, identificado na presente figura com (1). Caso o filtro seja criado incorretamente, quando é detetada a seleção do campo curso, é facultado um botão que permite limpar os filtros e apenas é possível selecionar o campo “Ano” se o campo “Curso” estiver preenchido o mesmo aplicando-se ao campo “Turma”.

Na mesma figura, e identificado com (2), temos um botão cuja função recai na exportação de um ficheiro Excel do horário que atualmente é criado manualmente. O calendário tem como auxílio botões de navegação, identificado com (3) e botões que permitem visualizar o calendário por mês, semana ou dia, identificado com (4).

Por fim, e identificado com (5), temos o calendário que permite criar, editar e eliminar horários com um número reduzido de cliques de uma maneira fácil e intuitiva

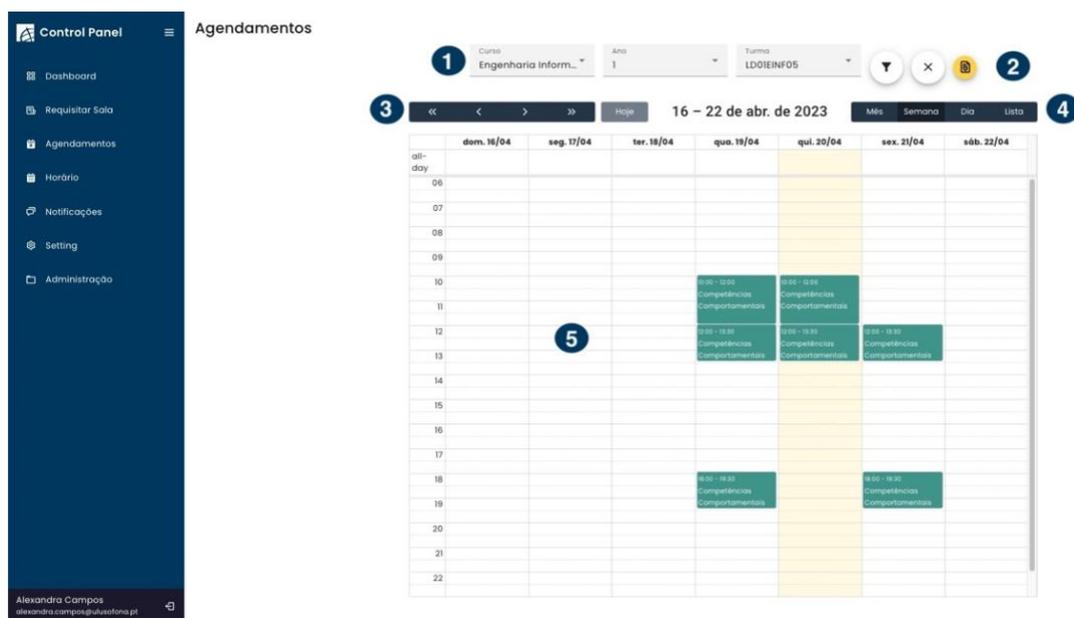


Figura 19 - Funcionalidades dos agendamentos no calendário

7.5 Abrangência

Este projeto teve a aplicação de vários conhecimentos, obtidos em unidades curriculares ao longo da licenciatura, como:

- Algoritmia e Estruturas de Dados, onde foram transmitidas técnicas de produção de programas de forma correta, de forma a garantir o adequado desempenho, o eficaz nível de tempo de execução e o eficiente consumo de memória, evitando assim erros causados pelo fraco desempenho das aplicações;
- Programação Web, onde aprendemos a criação e desenvolvimento de aplicações Web (*back-end* e *front-end*), providenciando uma visão global do desenvolvimento da Web;

- Engenharia de Requisitos e Testes, que foi uma unidade curricular com grande utilidade, permitindo entender o que é a gestão de sistemas de informação. O porquê de ser fundamental definir requisitos. Saber desenvolver e gerenciar uma metodologia de definição e estrutura de requisitos. Como alinhar os testes com os requisitos, assim como a importância de modelar os mesmos. Houve também uma passagem de conhecimento sobre fundamentos de gestão de projetos;
- Base de Dados, que teve por objetivo apresentar a importância da gestão de dados no seio de uma empresa, de forma a garantir quer a operação diária de negócio, bem como o apoio à decisão estratégica;
- Interação Humano-Máquina, que teve como principal objetivo a aquisição de conhecimento sobre as várias filosofias utilizadas ao longo dos tempos no desenho das interfaces homem-máquina;
- Engenharia de Software, onde é ensinado o desenho, escrita de código e testes, e manutenção de programas que têm um propósito de resolver problemas complexos com soluções eficientes e seguras. As técnicas de desenhar e desenvolver software de uma forma eficiente, legível, modular e escalável também tiveram bastante presentes.

Deste modo, pudemos então construir e implementar um software completo que satisfizesse as nossas necessidades e objetivos desejados.

7.6 Código Fonte e Demonstração

O respetivo código fonte da ULHT Control Panel (*back-end* e *front-end*) pode ser consultado através dos seguintes repositórios GitHub respetivamente:

- <https://github.com/DEISI-ULHT-TFC-2022-23/TFC-DEISI278-ULHT-control-panel-back-end>
- <https://github.com/DEISI-ULHT-TFC-2022-23/TFC-DEISI278-ULHT-control-panel-front-end>

A demonstração do funcionamento do ambiente de desenvolvimento pode ser visualizada através plataforma *Youtube* com o seguinte *link*:

<https://youtu.be/TQIKyCtJyTA>

8 Método e Planeamento

Utilizando a aplicação *YouTrack*, da empresa de desenvolvimento de software *JetBrains*, conseguimos organizar e controlar as implementações planeadas no *brainstorming* inicial, ao longo do projeto em si, fazendo com que a proposta cumpra de forma rigorosa as datas previstas.

Decidimos dividir o processo de desenvolvimento do projeto em 6 etapas [Figura 19]:

1. Criação de ambientes de desenvolvimento
2. Estruturação da base de dados
3. Desenvolvimento das *Applications Programming Interface – APIs (back-end)*
4. Desenvolvimento das páginas (*front-end*)
5. Período de testes
6. Correções e melhorias



Figura 20 - Cronograma proposto em formato *Gantt*

Desenvolvemos a *timeline* [Figura 20] tendo em conta o prazo estipulado, reservando algumas semanas antes do prazo de entrega para, se necessário, sejam implementadas melhorias ou realizadas correções essenciais para o bom funcionamento da aplicação.

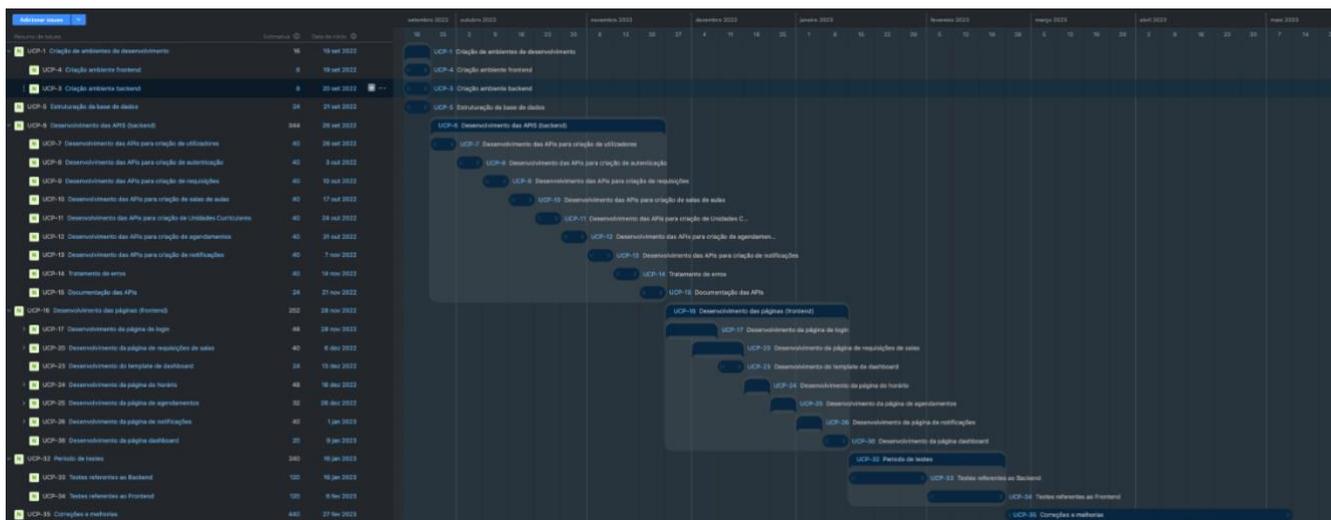


Figura 21 - Cronograma proposto em formato Gantt detalhado

1. Criação de ambientes de desenvolvimento

Etapa focada na criação dos ambientes de *front-end* e *back-end* utilizados para o desenvolvimento da aplicação web.

2. Estruturação da base de dados

Após a criação dos ambientes de desenvolvimento será estruturada uma base de dados que conterà os dados utilizados e apresentados na aplicação.

3. Desenvolvimento das APIs (*back-end*)

Esta etapa é destinada ao desenvolvimento das *applications programming interfaces*, que terão como função o tratamento dos dados fornecidos pelo utilizador e o envio dos mesmos armazenados na base de dados.

4. Desenvolvimento das páginas (*front-end*)

Nesta etapa serão desenvolvidas páginas e serão mostrados ao utilizador todos os dados necessários, através das APIs, para as funcionalidades pretendidas na aplicação.

5. Período de testes

O período de testes terá como finalidade, a realização de testes, relacionados com requisições às APIs, e a realização de testes focados na experiência do utilizador (*front-end*).

6. Correções e melhorias

Decidimos deixar o restante espaço de tempo até à entrega do projeto para a realização de correções necessárias devido a erros detetados no período de testes e para desenvolvermos possíveis novas funcionalidades que possam acrescentar valor à aplicação.

As tarefas apresentam-se bastante desenvolvidas tendo conseguido respeitar o planeamento, com a finalidade de se conseguir concluir todos os objetivos proposto à data de entrega final [Figura 21].

9 Plano de testes e validação

Sendo que a aplicação será para elementos da coordenação do DEISI e secretariado, serão implementados testes planeados [Tabela 6], que serão executados pela Prof. Dra Alexandra Campos, Assessora da Direção do DEISI e Coordenadora Pedagógica, responsável pela criação dos horários no início de cada semestre e por Lurdes Oliveira responsável pelo Serviço de Apoio Técnico-Administrativo (SATA).

Para tal serão realizados os seguintes testes:

Título	Descrição	Requisitos	Resultado Esperado
Criação de um horário	O utilizador preenche um formulário de criação de um horário.	RF23, RF24, RF25, RF26, RF28, RF29	O utilizador preenche um formulário de criação de um horário com os respetivos dados.
Filtrar por turma	O utilizador procede ao preenchimento do filtro para selecionar uma turma.	RF11, RF12, RF13, RF15, RF17	O utilizador seleciona o respetivo ano, curso e turma.
Eliminação de um horário	O utilizador realiza a eliminação de um horário.	RF22	O utilizador seleciona horário que pretende eliminar e após aparecer o <i>pop-up</i> clica no botão “Eliminar”.
Edição de um horário	O utilizador edita o horário que criou anteriormente.	RF22, RF30	O utilizador seleciona o horário que pretende editar e após aparecer o <i>pop-up</i> clica no botão “Editar”. Edita os campos que pretende modificar e clica no botão “Editar”
Navegação pelo calendário	O utilizador percorre o calendário de modo que desejar (dia, semana e mês).	RF16, RF19, RF20, RF21, RF22	O utilizador clica nos respetivos botões de navegação do calendário.

Login	O utilizador preenche o formulário de login e acedendo à aplicação com sucesso.	RF1, RF2, RF3, RF4	O utilizador preenche os campos do formulário com as suas credenciais.
Requisição de sala	O utilizador preenche o formulário de requisição de sala ao aceder à página de requisitar sala.	RF5, RF9, RF10	O utilizador preenche os campos do formulário de requisição de sala.
Submissão de notificações	O utilizador cria uma notificação e envia individualmente ou em grupo.	RF6, RF7, RF8	O utilizador acede à página de notificações e procede à criação de uma notificação.
Gerar ficheiro Excel	O utilizador gera um ficheiro Excel.	RF14	O utilizador cria um ficheiro Excel selecionando o botão amarelo depois de criado o respetivo horário
Administrar	O utilizador acede à página da administração para editar, criar ou eliminar dados.	RF31, RF32, RF33, RF34, RF35, RF36, RF37, RF38, RF39, RF40	O utilizador pretende administrar diversos dados na página de administração.

Tabela 6 - Tabela de testes

10 Resultados

10.1 Update: 27 de janeiro de 2023

O desenvolvimento do projeto apresenta-se com alguns avanços comparativamente à entrega intercalar. A página *login* [Figura 22] está finalizada, a componente do *front-end* foi desenvolvido e já se encontra a consumir a API de *login* e autenticação. Nesta página o utilizador tem a possibilidade de efetuar o *login* ou então, se necessário, recuperar a senha.

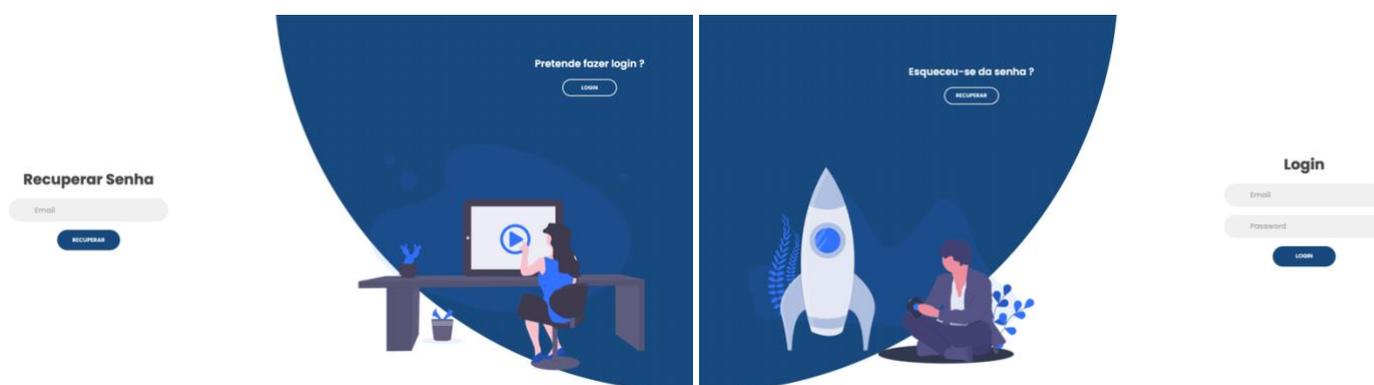


Figura 22 - Login e Recupera Senha

Após o utilizador efetuar o *login*, atualmente é redirecionado para a página do calendário que ainda se encontra em desenvolvimento. Temos como ambição desenvolver uma página de *dashboard* para onde o utilizador deverá ser redirecionado após efetuar o *login*.

Através das imagens [Figura 23 e 24] conseguimos visualizar os diversos estados da página do calendário, ou seja, o calendário tem ao seu dispor os filtros (mês, semana ou dia) que permitem ao utilizador ter várias possibilidades de visualização dependendo da tarefa que pretenda realizar.

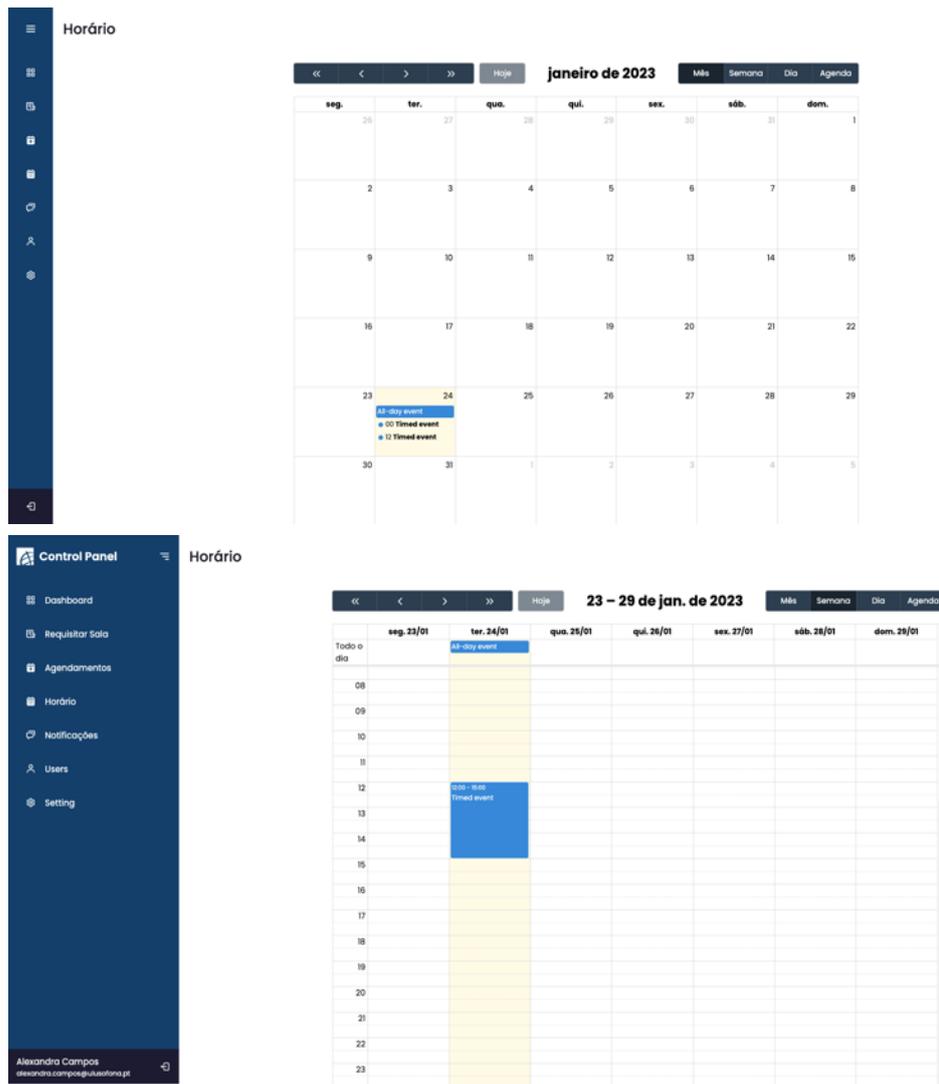


Figura 23 - Visualização do calendário (Mês e Semana)

Através da figura [Figura 23] podemos visualizar os dois estados do menu lateral, ou seja, foi desenvolvido um menu dinâmico, que permite ser escondido para um melhor aproveitamento do espaço livre do ecrã.

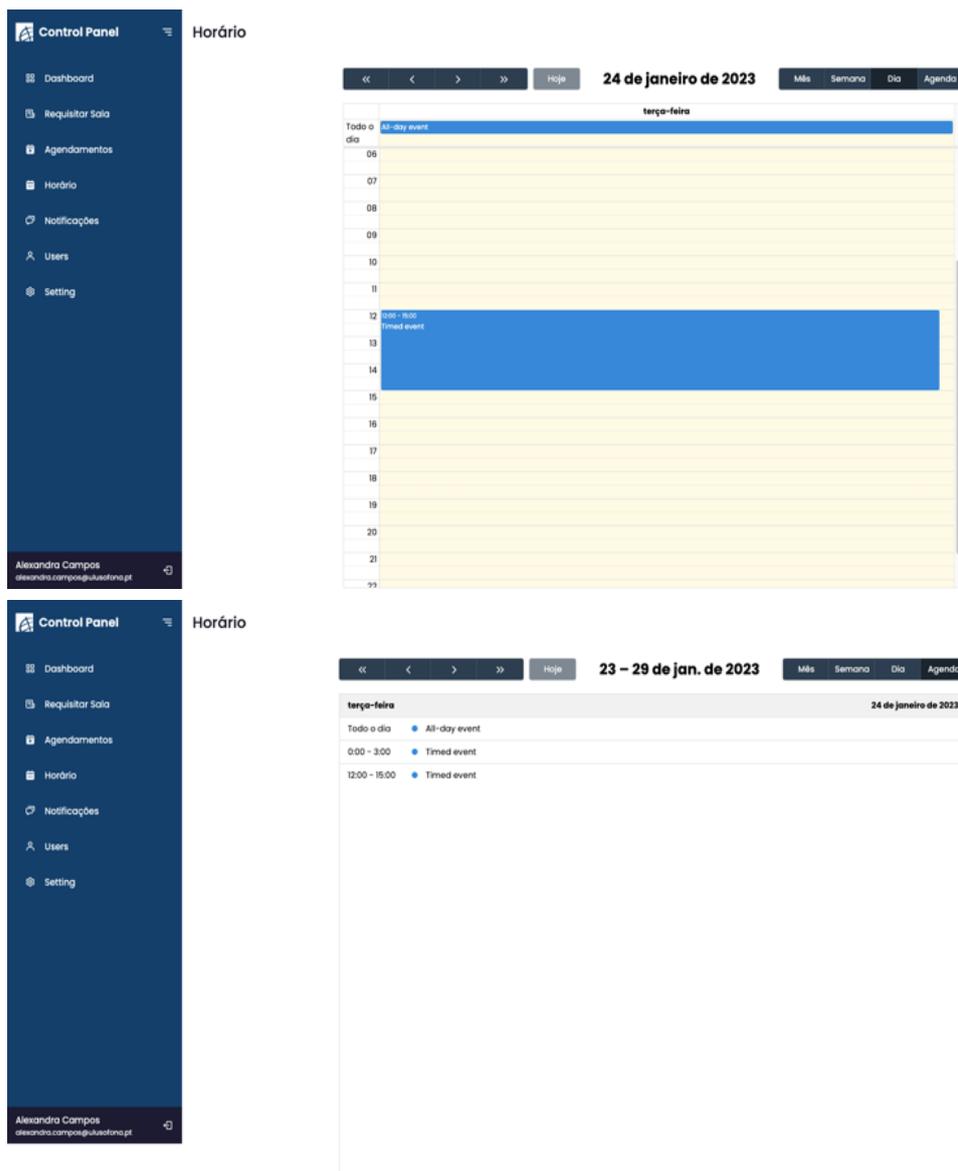


Figura 24 - Visualização do calendário (Dia e Agenda)

A página de requisição de salas encontra-se em desenvolvimento, pois inicialmente a página foi desenvolvida com o intuito de ser utilizada pelos alunos, mas neste momento está a ser adaptada para os docentes, pois são estes a nossa prioridade. Podemos então visualizar o ponto de situação da página através da seguinte figura [Figura 25].

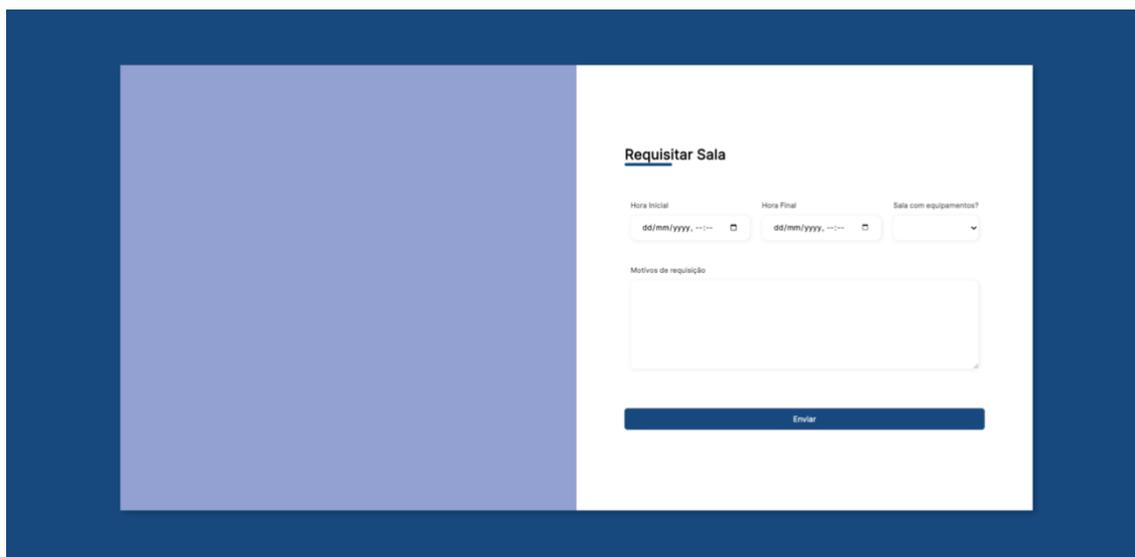


Figura 25 - Página de requisição de salas

10.2 Update: 30 de março de 2023

À data de 30 de março de 2023, a solução encontra-se em constante evolução, e de forma bastante positiva. A página de visualização do calendário [anexado na secção 7.4 *Melhorias de Usabilidade*] foi melhorada.

Através da página de visualização do calendário [Figura 26], para além de conseguimos visualizar os diversos estados da página do calendário feitos na entrega passada, também foi criada a opção de adicionar as unidades curriculares pretendidas com o auxílio de um *pop-up* [Figura 27] que permitem preencher com os dados pretendidos e criar o horário, assim como editá-lo ou eliminá-lo se o utilizador pretender.

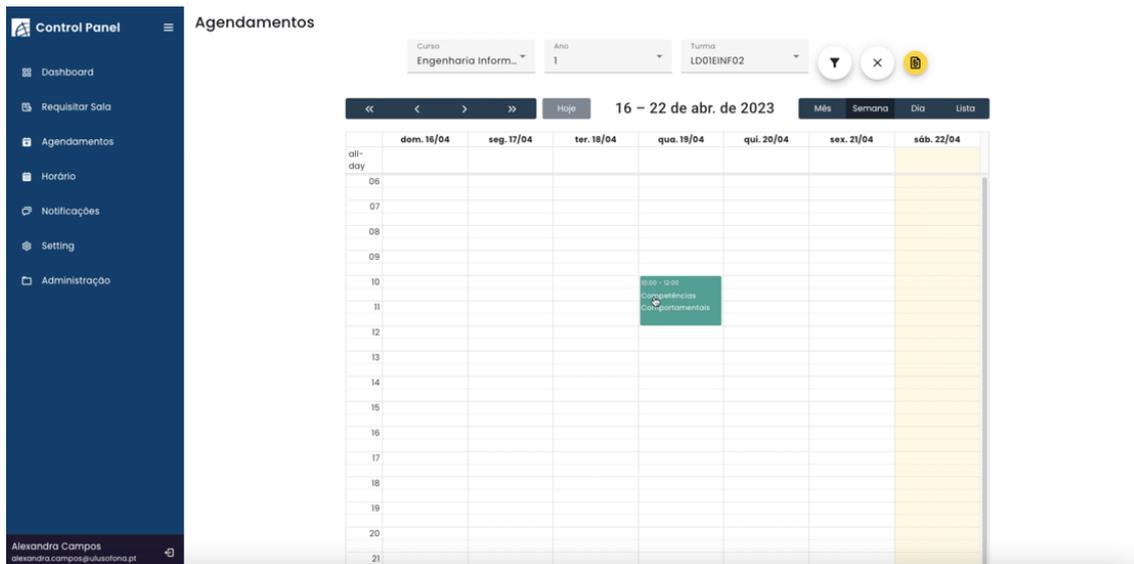


Figura 26 - Visualização do calendário

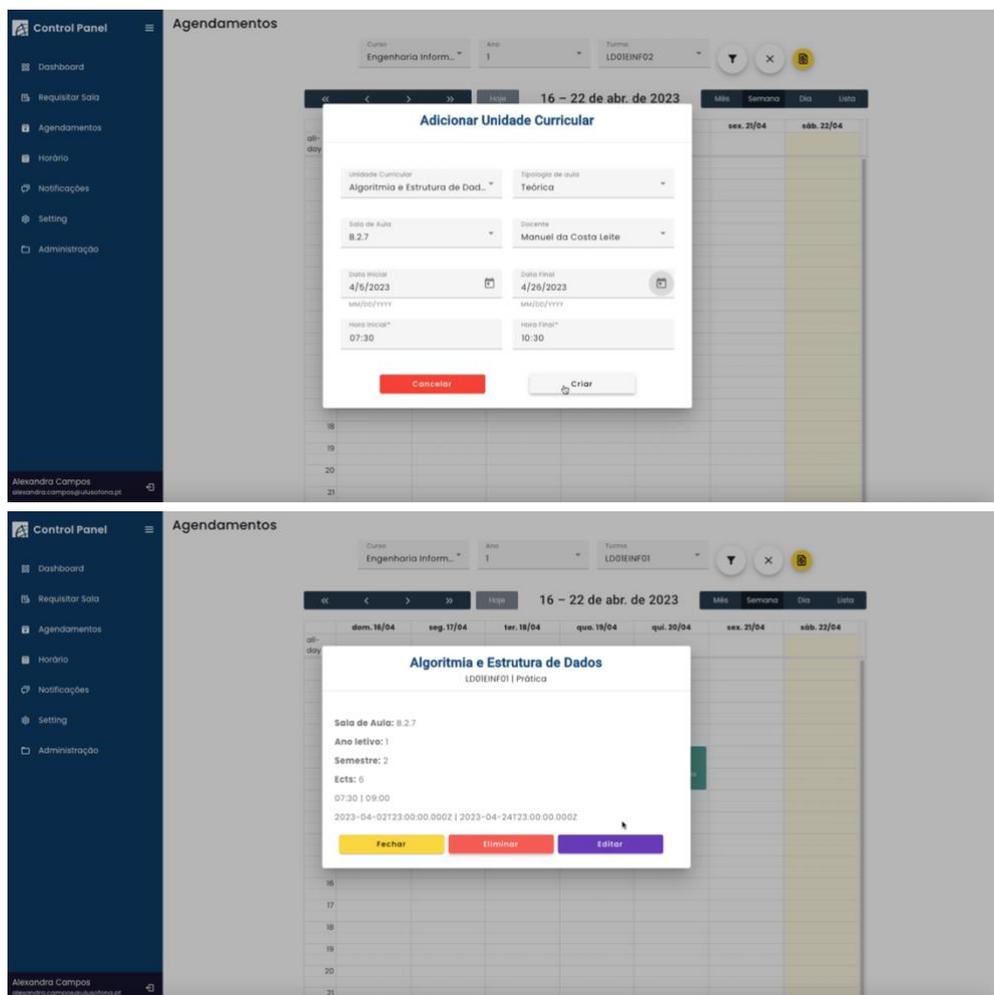


Figura 27 - Pop-up de Criação, Edição e Eliminação

Na página de administração [Figura 28] é possível, adicionar, editar e eliminar dados da base de dados úteis para a aplicação.

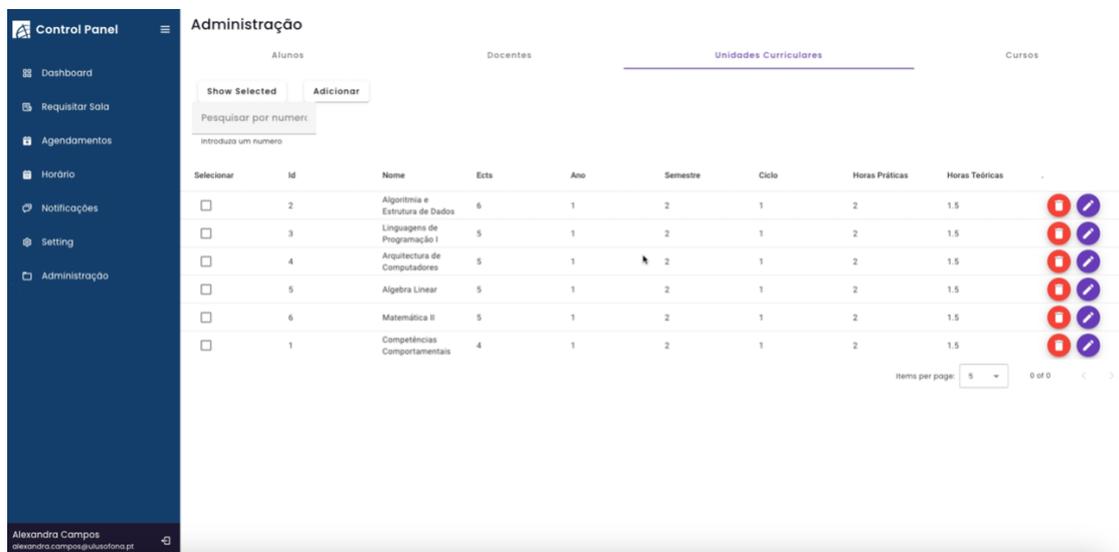


Figura 28 - Página de administração

10.3 Update: 25 de junho de 2023

À data de 29 de junho de 2023, a solução final encontra-se em significativamente bastante evoluída. Através da página do agendamento [Figura 29] os elementos de secretariado conseguem criar, editar e eliminar qualquer horário. Relativamente aos alunos e docentes, estes apenas conseguem visualizar horários e respetivas informações dos mesmos.

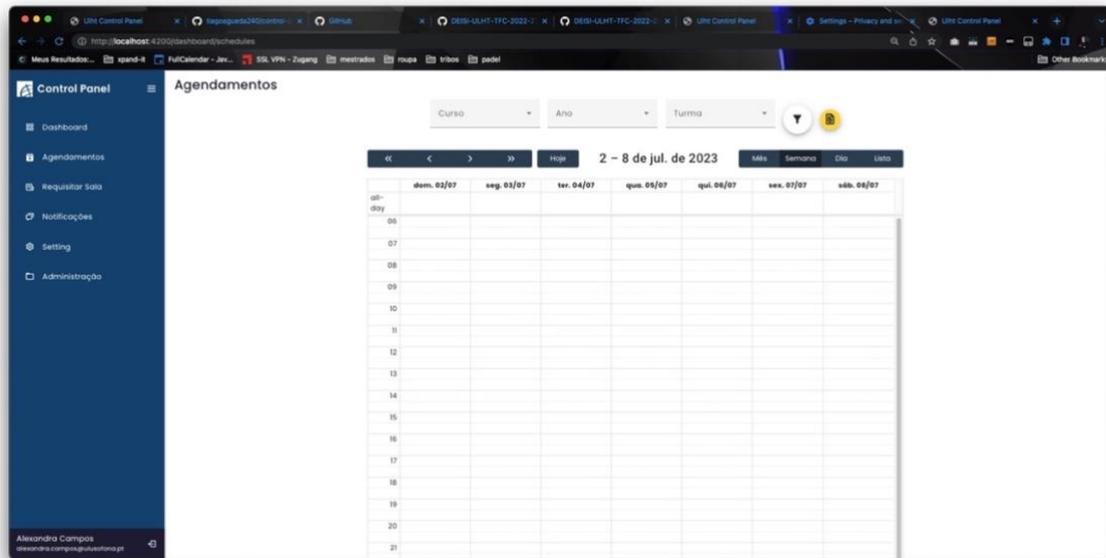


Figura 29 – Página do agendamento

A página de requisição de salas [Figura 30] sofreu um melhoramento significativo em termos de *layout*, tendo agora um *layout* mais *clean*.

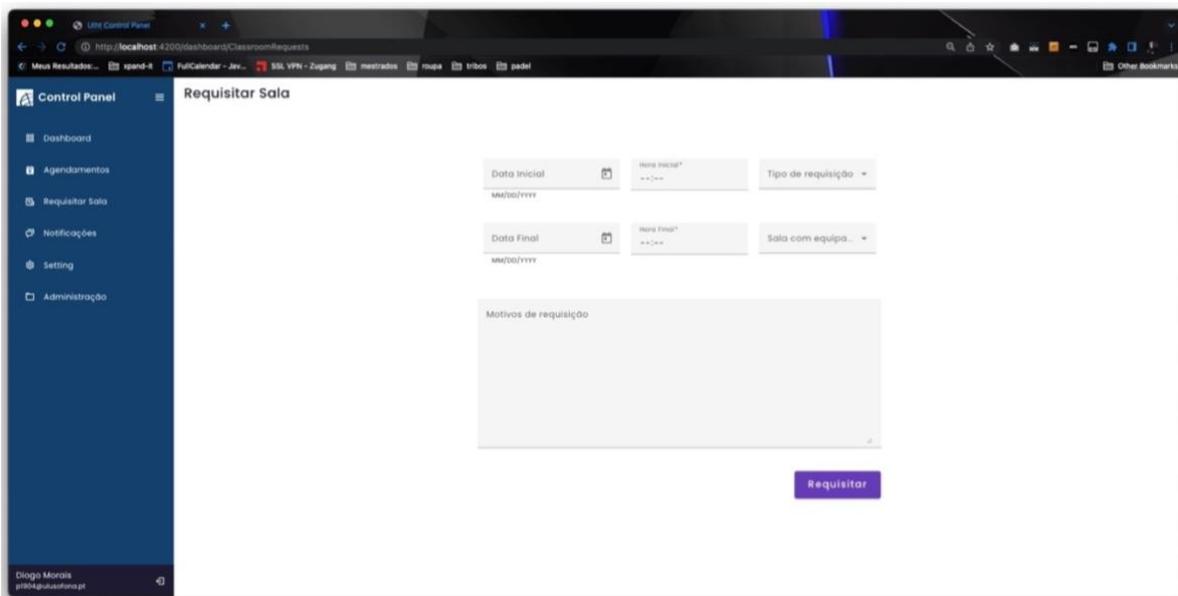


Figura 30 – Página de requisição de salas

Na página de notificações, os elementos de secretariado [Figura 31] têm a permissão para aprovar ou recusar requisições de sala realizadas pelos alunos e docentes.

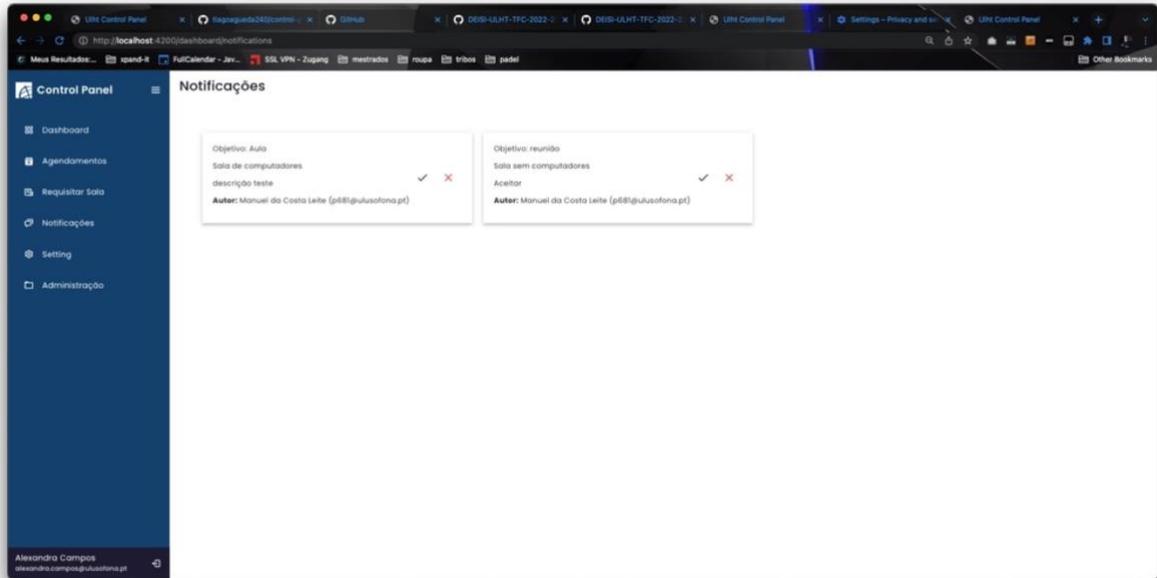


Figura 31 – Página de notificações (elementos do secretariado)

Ao aprovar uma requisição de sala [Figura 32], os elementos do secretariado necessitam de selecionar uma sala da lista de salas disponíveis de acordo com os requisitos e com as necessidades da requisição.

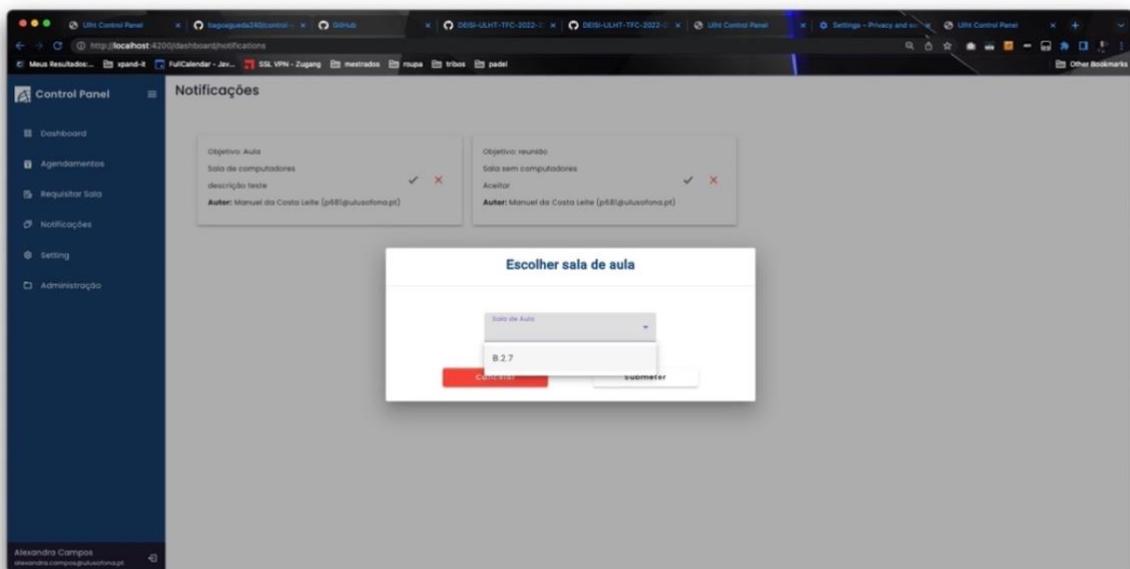


Figura 32 – Página de notificações (aprovação de requisição)

A página de notificações do utilizador [Figura 33] é possível visualizar o estado do pedido solicitado, como “Aprovado”, “Recusado” ou “Em Análise”.

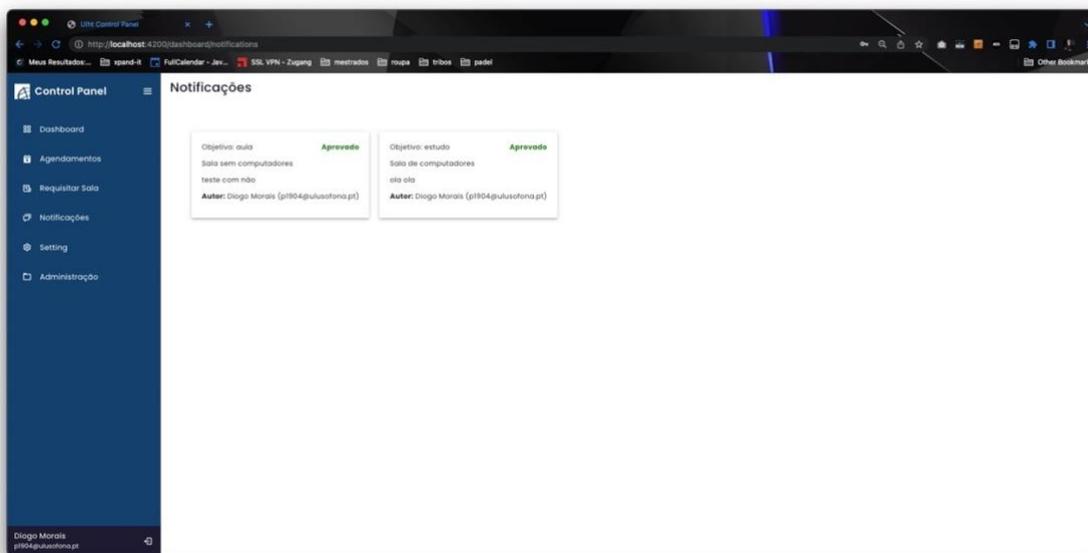


Figura 33 – Página de notificações (utilizador)

Na página da administração [Figura 34 e 35] é permitindo adicionar, editar e eliminar, alunos, docentes, unidades curriculares, cursos ou salas de aula.

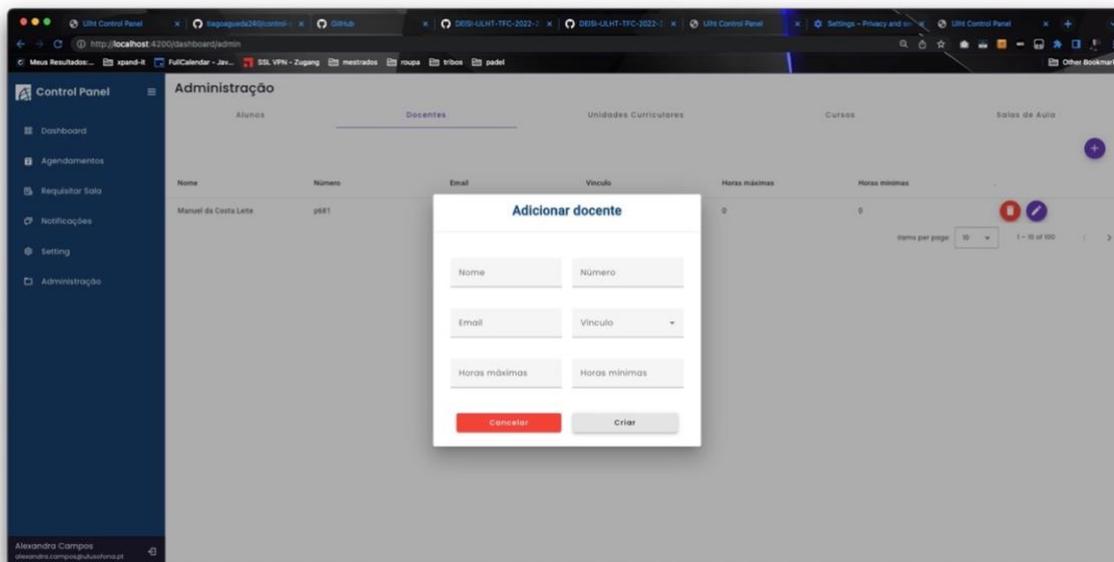


Figura 34 – Página da administração

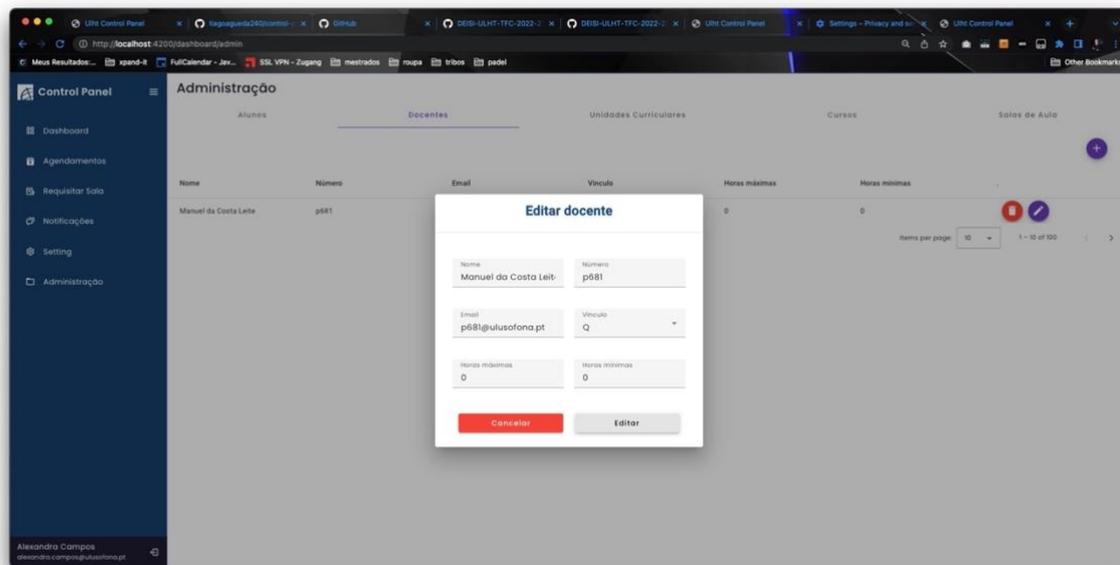


Figura 35 – Página da administração (Editar)

11 Conclusão

A produção manual de horários requer um esforço extenuante devido à necessidade de gerir os múltiplos conflitos de limitações impostos aos recursos a serem utilizados.

O ULHT Control Panel é uma ferramenta valiosa para aumentar a eficiência e a produtividade dos elementos do secretariado e docentes, sendo que elimina etapas de trabalho que têm de executar para chegar ao mesmo resultado. A aplicação permite a criação de horários, a atribuição de unidades curriculares, visualização de conflitos de horários e a reserva de salas, de uma maneira fácil e intuitiva. Além disso, com acesso à aplicação através de qualquer *browser*, os utilizadores podem gerir e criar os horários a partir de qualquer lugar e a qualquer momento. A interface é amigável, fácil de usar, e a capacidade de gerir várias tarefas simultaneamente, torna esta aplicação uma escolha ideal para a Universidade Lusófona.

11.1 Trabalhos Futuros

O foco do ULHT Control Panel centrou-se na criação de uma ferramenta para ajudar a coordenação do DEISI na preparação dos horários de cada ano letivo, assim como simplificar o processo de requisição de salas atual.

Considerando que os requisitos propostos neste documento serão finalizados com sucesso, os próximos passos a considerar para uma possível continuação deste Trabalho Final de Curso seriam a criação da página relativa ao *dashboard* onde poderá ser alocada várias informações importantes da aplicação e será a página que o utilizador será redirecionado após efetuar o *login*. As *settings* também será uma sugestão de uma possível implementação futura onde uma das funcionalidades poderá ser a alteração da *password* caso o utilizador se esqueça da mesma, entre outras opções e funcionalidades entregues à imaginação de quem quiser continuar com este projeto.

Outra funcionalidade que poderá ser útil será a criação de um espaço próprio para o agendamento e organização das mentorias.

12 Bibliografia

- [1] Nest.js [Consult. 2022-10]. Disponível em <https://docs.nestjs.com/>
- [2] Angular [Consult. 2022-10]. Disponível em <https://angularjs.org/>
- [3] Prisma [Consul. 2022-11]. Disponível em <https://www.prisma.io/docs>
- [4] Youtrack [Consult. 2022-11]. Disponível em <https://www.jetbrains.com/youtrack/>
- [5] Untis [Consult. 2022-11]. Disponível em <https://www.untis.at/en/solutions/use-untis-for/timetables-lessons-teacher-substitutions-1>
- [6] GOP – IST [Consult. 2022-11]. Disponível em <https://gop.tecnico.ulisboa.pt/>
- [7] Angular Material [Consult. 2022-11]. Disponível em <https://material.angular.io/components/icon/overview>
- [8] Swagger.io [Consult. 2023-05]. Disponível em <https://swagger.io/>

13 Anexos

13.1 Formulários

- Formulário disponibilizados aos elementos da coordenação do departamento DEISI: <https://forms.gle/hHsAGFPDY2wbzFFF7>
- Formulário disponibilizados a docentes do DEISI: <https://forms.gle/NonV8geTod2BpHWeA>
- Formulários disponibilizados a alunos do DEISI: <https://forms.gle/vy6LGwYnT9a31G1J9>

13.2 Progresso de trabalho

ID da issue	Tarefa	Estado	Dias despendidos
UCP - 22	Consumo de API de requests	Concluída	2
UCP - 21	Criação da componente requests	Concluída	3
UCP - 18	Criação da componente login	Concluída	4
UCP - 19	Consumo a API de autenticação	Concluída	2
UCP - 12	Desenvolvimento das APIs para criação de agendamentos	Concluída	5
UCP - 11	Desenvolvimento das APIs para criação de Unidades Curriculares	Concluída	5
UCP - 10	Desenvolvimento das APIs para criação de salas de aulas	Concluída	5
UCP - 9	Desenvolvimento das APIs para criação de requisições	Concluída	5
UCP - 8	Desenvolvimento das APIs para criação de autenticação	Concluída	5
UCP - 7	Desenvolvimento das APIs para criação de utilizadores	Concluída	5
UCP - 5	Estrutura da base de dados	Concluída	3
UCP - 4	Criação do ambiente front-end	Concluída	1
UCP - 36	Desenvolvimento da página dashboard	Aberta	3
UCP - 27	Criação da componente timetable	Concluída	3
UCP - 28	Consumo da API de horários	Concluída	3
UCP - 29	Criação da componente schedules	Concluída	4
UCP - 13	Desenvolvimento das APIs para criação de notificações	Concluída	3
UCP - 3	Criação ambiente back-end	Concluída	1
UCP - 35	Correções e melhorias	Concluída	55
UCP - 34	Testes referentes ao front-end	Concluída	15
UCP - 33	Testes referentes ao back-end	Concluída	15
UCP - 31	Consumo da API de notificações	Concluída	1
UCP - 30	Criação da componente notifications	Concluída	3
UCP - 15	Desenvolvimento do template de dashboard	Aberta	3
UCP - 14	Documentação das APIs	Concluída	3
UCP - 30	Tratamento de erros	Concluída	5

Figura 36 - Progresso de trabalho

13.3 Documentação dos serviços

A documentação da ULHT Control Panel [Figura 30] é um guia abrangente que fornece informações detalhadas sobre os serviços e *endpoints* disponíveis na API do ULHT Control Panel. Essa documentação permite que os desenvolvedores compreendam e utilizem efetivamente os recursos oferecidos pela API. O ficheiro pode ser visualizado através do seguinte *link*: https://github.com/tiagoagueda240/control-panel-frontend/blob/main/Swagger_UI.pdf

23/06/2023, 19:17 Swagger UI



Documentação dos serviços - ULHT Control Panel 1.0 OAS3

A documentação da ULHT Control Panel é um guia abrangente que fornece informações detalhadas sobre os serviços e endpoints disponíveis na API do ULHT Control Panel. Essa documentação permite que os desenvolvedores compreendam e utilizem efetivamente os recursos oferecidos pela API.

Utilizadores

POST /users

Parameters Try it out

No parameters

Request body required application/json

Example Value Schema

```
{
  "name": "Diogo Morais",
  "number": "p3465",
  "email": "p1904@ulusofona.pt",
  "password": "123abc!!ABC",
  "functionType": "professor",
  "vinculo": "Q",
  "maxHours": "0.0",
  "minHours": "0.0"
}
```

Responses

Code	Description	Links
200	Sucess	No links

localhost:3000/api#/Informações Curso/CourseController_remove 1/31

Figura 37 - Exemplo da documentação do serviço (POST)

14 Glossário

TFC	Trabalho Final de Curso
DEISI	Departamento de Engenharia Informática e Sistemas de Informação
CSV	Comma-separated values
PUI	Ponto Único de Informação
GOP	Gabinete de Organização Pedagógica
IST	Instituto Superior Técnico
RGPD	Regulamento Geral sobre a Proteção de Dados
API	Applications Programming Interface
CoC	Convention Over Configuration
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
SATA	Serviço de Apoio Técnico-Administrativo