



**Universidade do Minho**  
Escola de Engenharia

# Interface Pessoa-Máquina

## Trabalho Prático – Fase II

Grupo 12

<https://www.figma.com/design/Ps4VLCpZ66bKq5p2ypzrDR/Projeto-IPM>

[https://github.com/UMinho-ENGINF-IPM/trabalho-pr-tico-gp25\\_12](https://github.com/UMinho-ENGINF-IPM/trabalho-pr-tico-gp25_12)

4 de maio de 2025



Ana Oliveira

A104437



Humberto Gomes

A104348



Inês Marques

A104263



Rafael Vilas Boas

A76350



Sara Lopes

A104179

## Resumo

No âmbito da segunda fase deste trabalho prático, foi implementada, com base em modelação prévia, uma interface de utilizador de um sistema para a gestão de horários de um curso universitário, utilizado tanto pelos alunos como pelo diretor de curso. Neste documento, apresenta-se a interface implementada em Vue [1] (e outras tecnologias) e o seu processo de implementação, bem como algumas correções feitas ao modelo da interface. Por falta de tempo, não foi possível fazer testes da aplicação com utilizadores reais, não sendo possível fazer uma reflexão muito aprofundada sobre a qualidade do produto final. No entanto, consideramos que a aplicação desenvolvida cumpriu os requisitos pedidos.

## 1 Objetivos

O objetivo do trabalho prático proposto na Unidade Curricular de Interface Pessoa-Máquina é a modelação e o desenvolvimento de uma aplicação Web para a gestão de horários de um curso universitário, utilizada tanto pelos alunos como pelo diretor de curso. Em particular, deseja-se que a aplicação permita aos alunos consultar o seu horário e fazer pedidos de troca de turno ao diretor de curso, e que o diretor de curso possa manualmente colocar alunos em turnos (para resolver limitações do algoritmo de colocação e aceder a pedidos de alunos) e aceder a pedidos de professores para mudança de turnos para salas maiores. Para mais detalhes, o enunciado do trabalho prático pode ser consultado nos anexos deste relatório. Idealmente, a aplicação deve ajudar o utilizador a não cometer erros, como colocar alunos em turnos que originem sobreposições de horário.

Na primeira fase deste trabalho, fez-se a modelação da aplicação em Figma [2], e nesta fase, procurou-se implementar a interface modelada em Vue [1]. Também se realizaram algumas alterações ao modelo, a corrigir alguns aspetos que se acharam relevantes.

## 2 Alterações ao Modelo da Interface

Na fase anterior deste trabalho prático, construiu-se, em Figma [2], um modelo da interface a implementar. Apesar de uma boa avaliação nesta fase, a docência da UC de Interface Pessoa-Máquina reparou em alguns aspetos que podiam ser melhorados. Nesta secção, apresentam-se as mudanças que foram feitas ao modelo da interface devido tanto aos comentários dos docentes como a outros aspetos que o nosso grupo de trabalho se apercebeu que podiam ser melhorados.

Em primeiro lugar, a página “Iniciar Sessão” não é ideal para a prevenção de erros: é possível

que o utilizador submeta parcialmente as suas credenciais (apenas o seu endereço eletrónico ou apenas a sua palavra-passe), e o sistema reagirá com um erro. Para prevenir este erro, deve ser impossível que um utilizador submeta as suas credenciais até as escrever todas. Logo, na nova versão do modelo da interface, o botão de submissão de credenciais encontra-se desativado até o utilizador escrever tanto o seu endereço eletrónico como a sua palavra-passe, como mostra a figura abaixo:

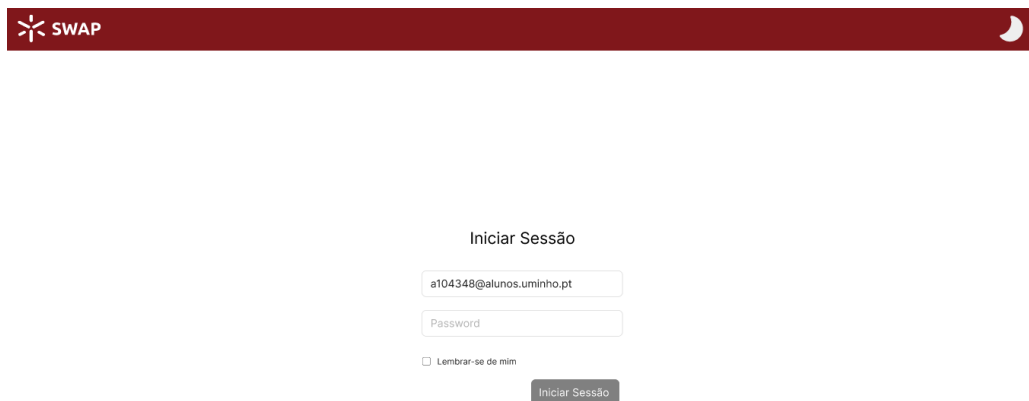


Figura 1: Captura de ecrã do protótipo da página “Iniciar Sessão” com botão desativado.

É importante que o utilizador saiba por que este botão se encontra desativado, pelo que, tal como foi feito em outros botões na interface, uma *tooltip* foi utilizada para justificar por que não é possível interagir com o botão:

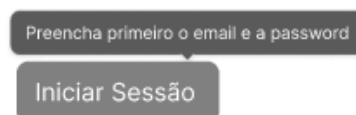


Figura 2: *Tooltip* sobre o botao de início de sessão desativado.

Ademais, na página “Resolver Problemas”, o título da página foi removido, e substituído por uma barra de pesquisa. Em primeiro lugar, o título da página não era necessário, visto que o utilizador já sabe em que página se encontra olhando para a hiperligação realçada na barra de navegação. Depois, o espaço que se ganha na barra lateral com a remoção deste título pode ser usado uma barra de pesquisa, um elemento muito útil para procurar alunos em grandes listas

de problemas (o cenário 1 aponta para 45 alunos sem turnos atribuídos).

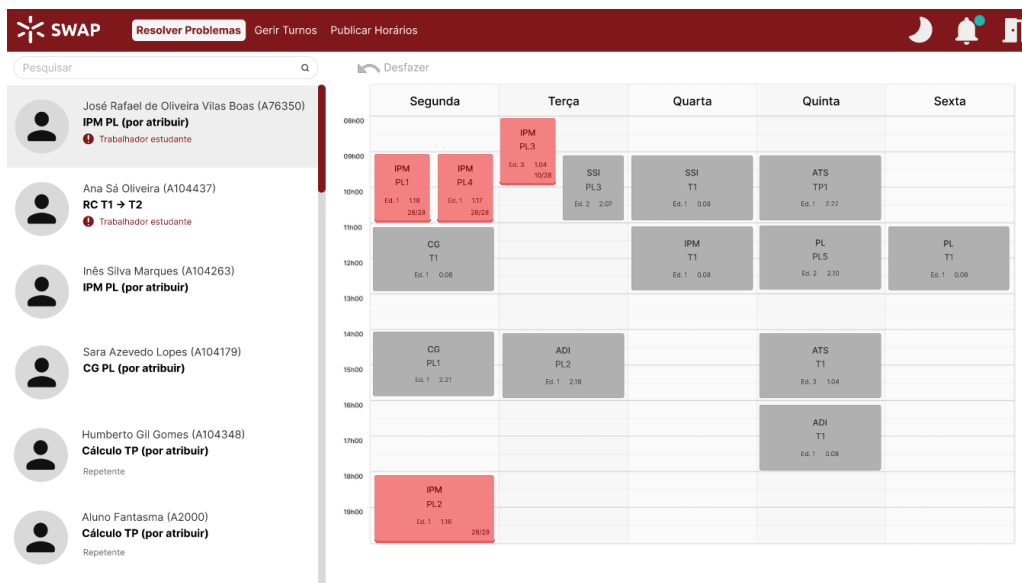


Figura 3: Captura de ecrã do protótipo da página “Resolver Problemas”.

De seguida, uma sugestão da docência de Interface Pessoa-Máquina foi a eliminação da página “Publicar Horários”, sendo estes atualizados sempre que o diretor de curso faz uma alteração. No entanto, não consideramos esta solução viável, visto que o diretor de curso pode desejar colocar os horários dos alunos em estados intermédios sem que estes os vejam. Por exemplo, pode desejar remover vários alunos dos seus turnos, para os adicionar a outros turnos da mesma UC, assim, por exemplo, abrindo vagas em turnos mais procurados. No entanto, reparou-se que é importante realçar quando as mudanças feitas pelo diretor de curso ainda não são públicas. Por este motivo, tal como foi feito para o ícone de notificações, quando há alterações de horário por publicar, um pequeno círculo é adicionado ao canto superior direito da hiperligação para a página “Publicar Horários”, como mostra a figura abaixo:



Figura 4: Captura da barra de navegação com realce na hiperligação para a página “Publicar Horários”.

No entanto, partindo da solução de não ser necessário publicar os horários, fez-se não ser necessário, na página “Gerir Turno”, guardar as alterações feitas a um turno: estas são guardadas à medida que o utilizador as faz. Logo, o botão “Guardar” foi substituído por um botão “Voltar”. Na mesma página, desfazer uma alteração deixou de ser feito através de uma *toast*

que aparecia sempre que uma alteração era feita, e sim por um botão “Desfazer”, de forma consistente com a página “Resolver Problemas”:

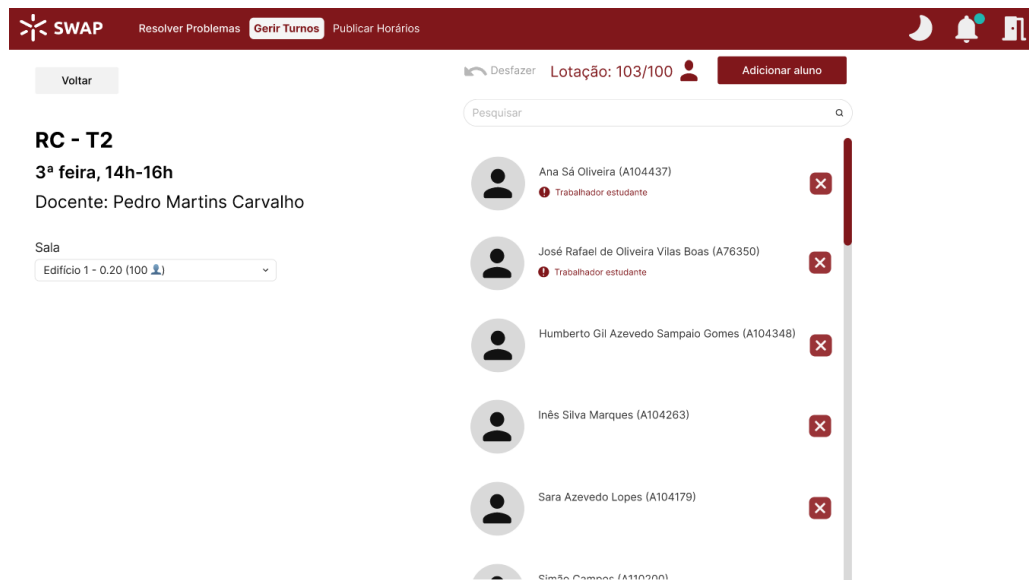


Figura 5: Captura de ecrã do protótipo da página “Gerir Turnos”.

Na página “Notificações”, o botão para alternar entre visualizar todas as notificações ou apenas as não lidas foi removido, visto que as mesmas já se encontram ordenadas por este critério:

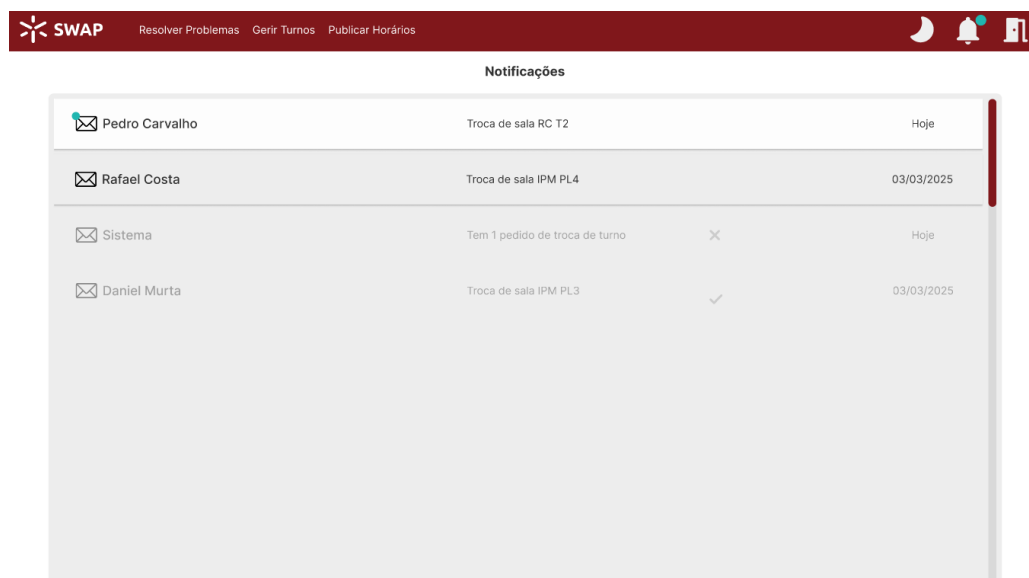


Figura 6: Captura de ecrã do protótipo da página “Notificações” do diretor de curso.

Por último, também se seguiu a sugestão da equipa docente de mostrar o estado dos pedidos

nas páginas “Histórico de Pedidos” e “Notificações do Diretor de Curso”. Abaixo, apresenta-se, como exemplo, o modelo da página “Histórico de Pedidos”. Apesar do uso de linguagem iconográfica, tal como no resto da aplicação, os ícones apresentados têm todos *tooltips*, que são ativadas após o cursor os sobrevoar por alguns segundos:



Figura 7: *Tooltip* sobre o estado de um pedido.

### 3 Componentes Implementados

O primeiro passo no processo de implementação da interface foi a identificação de componentes e respetiva implementação. Foram identificados como componentes *widgets* comuns (botões, campos de texto, ...) e outros elementos presentes em várias páginas / várias vezes na mesma página. Note-se que *widgets* simples foram implementados como componentes (em vez de serem usados diretamente) para haver um maior controlo sobre o estilo que é aplicado aos mesmos, e para ser possível suportar várias variantes dos mesmos. Segue-se uma breve lista dos vários componentes desenvolvidos:

#### 3.1 Alerta

Este componente estático apresenta uma informação importante ao utilizador. É utilizado, por exemplo, para informar o utilizador na página “Iniciar Sessão” que as credenciais que submeteu se encontram incorretas, ou nos componentes “Aluno” e “Problema” que um aluno é trabalhador-estudante, como se vê na figura 14.

#### 3.2 Barra de Navegação

Este componente está presente no topo de todas as páginas implementadas, e permite ao utilizador rapidamente navegar para outras páginas de interesse, bem como alterar o seu tema (claro ou escuro) e terminar sessão. Este componente, como se vê abaixo, existe em três variantes,

uma para antes de se ter iniciado sessão, uma para alunos, e outra para o diretor de curso. A barra de navegação subdivide-se em vários subcomponentes, que não serão mencionados neste relatório, visto que apenas existem para dividir a barra de navegação em partes mais pequenas, dando origem a código mais modular.

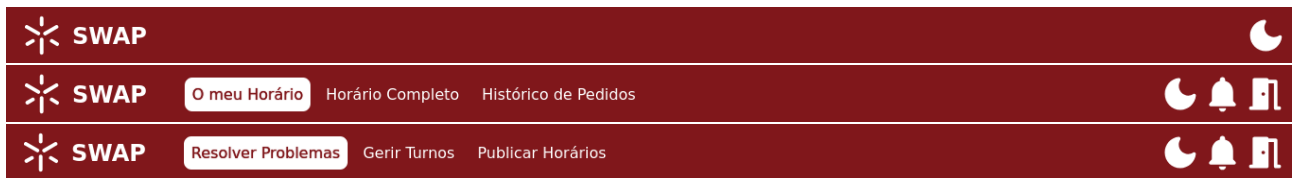


Figura 8: De cima para baixo, variantes de *login*, de aluno, e do diretor de curso da barra de navegação.

### 3.3 Botão

Este componente, presente em diversas páginas da aplicação, trata-se de um *widget* no qual se pode clicar para se realizar uma ação pré-determinada. Existem várias variantes de botão (ativo, de cancelamento, e desativado), distinguíveis pela sua cor. Adicionalmente, botões desativados podem ter uma *tooltip* a explicar por que motivo não é possível interagir com eles. Abaixo, apresentam-se os três tipos de botão implementados:

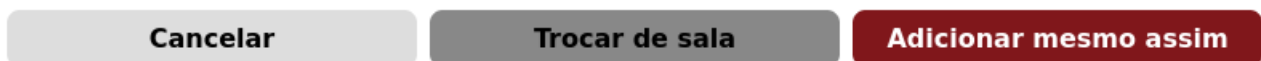


Figura 9: Da esquerda para a direita, variantes de cancelamento, desativada e ativa de botões.

### 3.4 Botão para Desfazer

Este componente trata-se de um botão especial, utilizado para desfazer ações nas páginas “Resolver Problemas” e “Gerir Turno”. Pode ser visto na figura abaixo:



Figura 10: Botão para desfazer.

### 3.5 Capacidade

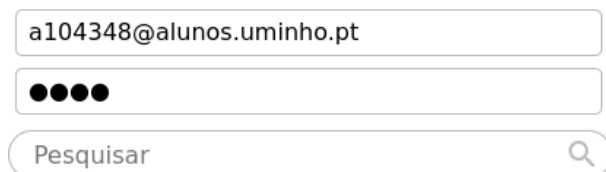
Este componente apresenta a utilização de um turno em função da sua capacidade, sendo utilizado na página “Gerir Turno”, no componente “Turno”, e no diálogo de informação sobre um turno nas páginas “O Meu Horário” e “Horário Completo”. Opcionalmente, este componente pode passar a ser representado a vermelho caso um turno exceda a sua capacidade, como pode ser visto na figura abaixo:

Lotação: 40 / 10 

Figura 11: Capacidade de um turno sobrelotado.

### 3.6 Campo de texto

Este componente, presente em várias páginas da aplicação, permite ao utilizador escrever texto a ser consumido pelo programa. Existem três variantes deste componente, para entrada de texto, para entrada de *passwords*, e uma barra de pesquisa. Na figura abaixo, podem observar-se estas três variantes do campo de texto:



The figure displays three distinct text input components stacked vertically. The top component is a standard text input field containing the email address 'a104348@alunos.uminho.pt'. The middle component is a password input field, represented by four black dots. The bottom component is a search bar with the placeholder text 'Pesquisar' and a magnifying glass icon on the right side.

Figura 12: Campos de texto para entrada textual, entrada de *passwords*, e barra de pesquisa.

### 3.7 *Check box*

Este componente permite ao utilizador definir o valor de um booleano. Destaca-se o seu uso no componente “Seletor de Turnos”, visível na figura 19.



### 3.8 Dropdown

Presente apenas na página “Gerir turno”, este componente permite ao utilizador escolher uma de diversas opções, no caso, uma de diversas salas, como se pode ver abaixo:



Figura 13: *Dropdown*.

### 3.9 Estudante e Problema

Estes dois componentes são bastante semelhantes, e permitem apresentar um estudante e um problema (pedido de troca ou turno por atribuir), respetivamente. O componente “Problema” pode observar-se abaixo:

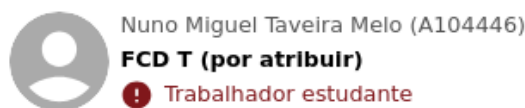


Figura 14: Problema de turno por atribuir. Caso apenas se apresentasse o aluno, a linha “FCD T (por atribuir)” não seria visível.

### 3.10 Horário

Este componente, utilizado em diversas páginas, permite apresentar um conjunto de turnos, organizados temporalmente ao longo de uma semana, como se pode ver abaixo:

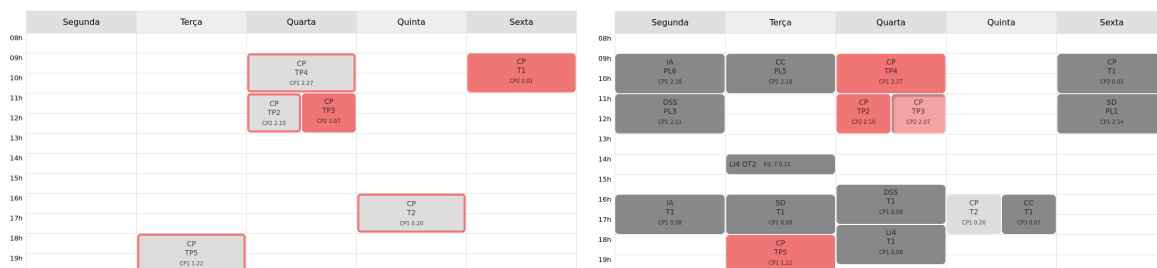


Figura 15: Horários com diferentes tipos de turnos.

### 3.11 Ícone da Aplicação

Este componente estático é usado na barra de navegação e na barra de janela do componente “Popup”. Pode ser visto na figura abaixo:



Figura 16: Ícone da aplicação.

### 3.12 Lista de Estudantes

Este componente, utilizado na página “Gerir Turno”, apresenta uma lista de estudantes, onde cada estudante se encontra associado a um botão, que pode ser usado para o adicionar a / remover de um turno. Este componente pode ser visto na figura abaixo:



Figura 17: Lista para remoção de estudantes.

### 3.13 *Popup*

Este componente é um diálogo com uma borda de janela. Várias páginas utilizam *popups*, mas dois destes, para mostrar informação de um turno e para confirmar a troca entre dois turnos,

apresentados abaixo, foram transformados em componentes, visto que estão presentes em mais do que uma página, no caso, “O meu Horário” e “Horário Completo”.



Figura 18: Diálogos de informação de turno e para confirmação de troca de turnos.

### 3.14 Seletor de turnos

Este componente, presente nas páginas “Horário Completo” e “Gerir Turnos”, permite ao utilizador escolher que turnos deseja ver no horário apresentado.

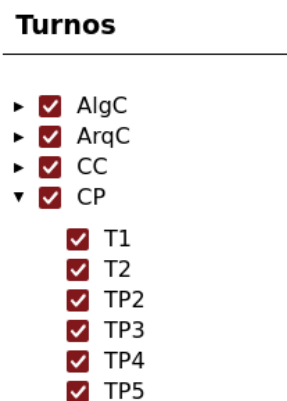


Figura 19: Seletor de turnos.

### 3.15 *Toast*

Este trata-se de um componente para apresentar um alerta ao utilizador, sem que este deixe de poder utilizar a página. É normalmente utilizado para informar o utilizador de sucesso nas operações que realizou: que um pedido de mudança de horário foi enviado nas páginas “O Meu Horário” e “Horário Completo”, e que os horários foram publicados com sucesso, na página “Resolver Problemas”. Abaixo, pode ver-se o exemplo de uma *toast*:



Figura 20: *Toast*.

### 3.16 Turno

Este componente, utilizado no componente “Horário” (figura 15), permite apresentar um turno. Existem várias variantes de turnos, necessárias para ser possível distinguir turnos com os quais é possível interagir, turnos pertencentes ao horário do aluno, o turno do qual um aluno requisita para sair, *etc.*. É também possível escolher se se deseja que um turno apresente ou não a sua capacidade.

### 3.17 Botão com ícone

Este componente trata-se de um *widget* no qual se pode clicar para se realizar uma ação, sendo utilizado no componente “Notificação”, do diretor de curso (figura 22). Existem várias variantes para as ações de rejeitar, aprovar ou navegar para a página de resolução de um pedido, distinguíveis pelo seu ícone. Abaixo podem ser vistos os três tipos do botão implementados:



Figura 21: Da esquerda para a direita, variantes de rejeição, aprovação e navegação dos botões.

### 3.18 Notificação

Este componente, utilizado no componente “Lista de Notificações” (figura 23), permite apresentar o conteúdo e estado de uma notificação / pedido (troca de turno, de sala, ou alerta). Existem várias variantes deste componente, apresentadas a diferentes utilizadores em diferentes cenários (notificação de aluno, do diretor de curso, ou pedido). Adicionalmente, notificações do diretor de curso, quando sobrevoadas pelo cursor, apresentam três botões, que permitem rejeitar, aprovar ou navegar para a página de resolução do pedido apresentado na notificação. As variantes do diretor de curso e de um pedido podem observar-se abaixo:

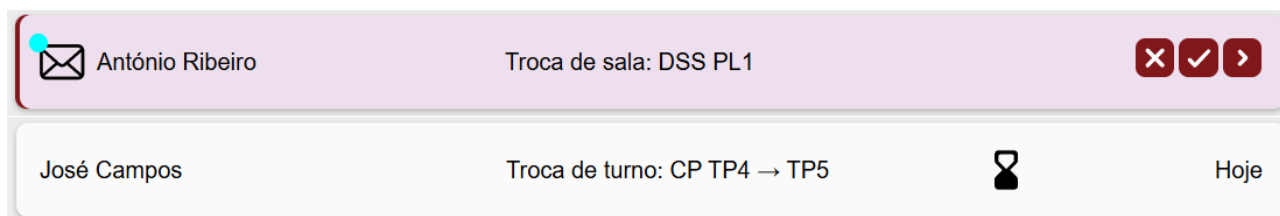


Figura 22: De cima para baixo, variantes notificação de diretor de curso e de pedido.

### 3.19 Lista de Notificações

Este componente, utilizado nas páginas de “Histórico de Pedidos” e “Notificações”, apresenta uma lista de notificações / pedidos, em que cada elemento é uma instância do componente “Notificação”. Permite listar as notificações / pedidos de um utilizador, organizados temporalmente e por estado do pedido, como se pode ver abaixo:



Figura 23: Lista de notificações, do diretor de curso.

## 4 Páginas Implementadas

Para não repetir o mesmo conteúdo duas vezes neste relatório, as páginas implementadas são apresentadas na secção “Manual de Utilização”.

## 5 Tecnologias Utilizadas

Para a realização deste projeto, foi necessário o uso de diversas tecnologias. Nesta secção, procura-se apresentar as diversas tecnologias utilizadas, como estas foram utilizadas, e as facilidades e dificuldades sentidas com cada uma.

### 5.1 HTML e CSS

Visto que a aplicação desenvolvida é uma aplicação Web, o uso de HTML e CSS é imperativo em quase todas as *frameworks*. De um modo geral, o desenvolvimento de componentes e páginas com recurso a estas tecnologias foi fácil, especialmente devido à sua natureza declarativa. Outra facilidade do uso destas tecnologias foram as *flexboxes*, que simplificaram a organização dos elementos nos componentes e nas páginas.

No entanto, houve algumas dificuldades em garantir que o código desenvolvido funcionava em todos os navegadores: como nem todos os navegadores suportavam todas as funcionalidades desejadas, foi necessário testar a aplicação desenvolvida em Chromium (Blink), Firefox (Gecko) e Safari (WebKit). Várias vezes, foi necessário adicionar regras de CSS com *vendor prefixes*, ou até refazer algumas funcionalidades devido a diferenças de funcionamento entre navegadores.

### 5.2 TypeScript

No projeto desenvolvido, foi utilizado TypeScript [3], linguagem que é transpilada para JavaScript, mas que permite alguma verificação de tipos. O TypeScript foi uma grande ajuda para garantir a correção do código da aplicação, e assegurar, em *compile-time*, que a aplicação não teria exceções em *runtime* devido a erros de tipos. Ademais, o compilador não obriga que o código seja tipado durante o processo de desenvolvimento, pelo que este pode continuar a ser bastante veloz. Assim, apenas quando se acaba de escrever um módulo / componente / página, podem adicionar-se anotações de tipo ao código para garantir a sua correção.

### 5.3 Vue

A *framework* utilizada para o desenvolvimento da aplicação foi Vue.js [1]. Esta *framework*, devido à sua natureza declarativa e reativa, foi de muito fácil atualização: alterar um objeto reativo conduzia à atualização automática das partes do componente / da página que sofreram

alterações. Por exemplo, ao contrário do Blazor [4], utilizado em Laboratórios de Informática IV, não era necessário chamar uma função (`StateHasChanged`) sempre que era necessário atualizar os conteúdos da DOM.

Apesar de poucas, foram sentidas algumas dificuldades no uso de Vue, especialmente quando era necessário integrar funcionalidades não reativas (no caso, a resposta a um evento de mudança de uma *media query*) com a reatividade do Vue, tendo sido necessários *hacks* para forçar a atualização dos objetos reativos.

## 5.4 Pinia

A biblioteca Pinia [5] foi utilizada para armazenamento do estado da aplicação, como as credenciais do utilizador, o tema da aplicação (claro ou escuro) e os turnos escolhidos nas páginas “Horário Completo” e “Gerir Turnos”, para que esta informação não desapareça quando se navega entre diferentes páginas da aplicação. Adicionalmente, o *plugin* `pinia-plugin-persistedstate` [6] foi utilizado para persistir algum estado no `localStorage` do navegador, para este ser mantido mesmo após o navegador ser fechado. Por exemplo, as credenciais do utilizador são armazenadas no `localStorage` quando este escolhe que deseja que a aplicação se “lembre de si”.

O Pinia, devido à sua integração com Vue, foi das ferramentas de mais fácil utilização, visto que os objetos armazenados nos *stores* do Pinia podiam ser utilizados como qualquer outro objeto reativo do Vue.

## 5.5 JSON Server

É necessário que a aplicação desenvolvida tenha dados que possa apresentar. Na UC de Interface Pessoa-Máquina, não é pedido que se implemente um *backend* completo para a aplicação, pelo que se utilizou `json-server` [7], um programa que cria uma API REST a partir dos dados num ficheiro JSON.

A utilização desta tecnologia foi complicada, não devido a complexidade da tecnologia em si, mas sim devido ao trabalho necessário para organizar os dados da API num formato adequado para apresentação. Devido às capacidades muito limitadas de interrogação do `json-server`, várias interrogações tiveram de ser implementadas imperativamente em JavaScript, ao contrário de declarativamente numa linguagem especializada, como se teria feito caso se tivesse utilizado uma base de dados. O uso de uma base de dados seria, por este motivo, uma grande melhoria

ao projeto.

## 5.6 Outras tecnologias

O `npm` [8] é o gestor de pacotes do NodeJS, que permite, no ficheiro `pacakage.json`, definir as dependências do projeto desenvolvido. Além disso, também é possível definir *scripts* como `npm run dev`, para abrir a aplicação com ferramentas de desenvolvimento, e `npm run preview`, para pré-visualização da aplicação final. Para melhorar a experiência de desenvolvimento, estes dois *scripts* foram modificados para, além de executarem o servidor HTTP da aplicação, também executarem o JSON Server. Também foram adicionados *scripts* para, com recurso às ferramentas ESLint [9] e Prettier [10], automaticamente fazer uma análise estática e formatação do código, respetivamente. Estes *scripts* são usados na *pipeline* de CI (*Continuous Integration*) do repositório do projeto (GitHub Actions), obrigando a que todos os *pull requests* tenham código correto e bem formatado.

A principal dificuldade do uso do `npm` foi garantir que os *scripts* criados funcionavam em diferentes sistemas operativos, visto que alguns comandos utilizados apenas estavam disponíveis, por exemplo, em Linux.

## 6 Manual de Utilização

### 6.1 Diretor de Curso

#### 6.1.1 Cenário 1

Neste cenário, o diretor de curso entra na aplicação, consulta a ocupação dos turnos, e aloca os alunos que o algoritmo de geração de horários automático não alocou. Finalmente, publica o horário dos alunos. Para isso, terá de realizar os seguintes passos:

**- Iniciar sessão com as suas credenciais:**

Preencher o ecrã seguinte com o seu endereço eletrónico e palavra-passe, terminando a clicar no botão “Iniciar Sessão”:



**Iniciar Sessão**

Email

Password

☐ Lembrar-se de mim

Iniciar Sessão

Figura 24: Página “Iniciar Sessão” por preencher.

**Iniciar Sessão**

jfo@di.uminho.pt

\*\*\*\*\*

☐ Lembrar-se de mim

Iniciar Sessão

Figura 25: Página “Iniciar Sessão” preenchida com credenciais.

#### - Ver a ocupação dos turnos:

A aplicação desenvolvida procura minimizar o esforço de memória necessário para a sua utilização. Não é necessário que o diretor de curso consulte a capacidade dos turnos para saber onde pode e não pode colocar alunos (ver página “Resolver Problemas” abaixo). Mesmo assim, pode fazê-lo, como se mostra nesta secção, primeiro acedendo à página “Gerir Turnos”:

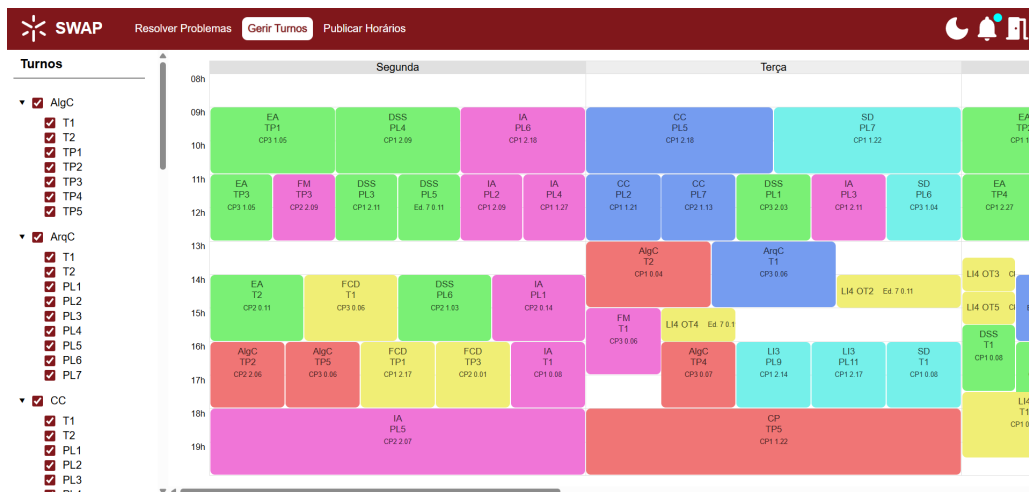


Figura 26: Página “Gerir Turnos”.

É possível, através da barra lateral, selecionar que turnos são apresentados no horário, caso alguns turnos ou UCs não sejam de grande interesse para a tarefa atual do diretor de curso:

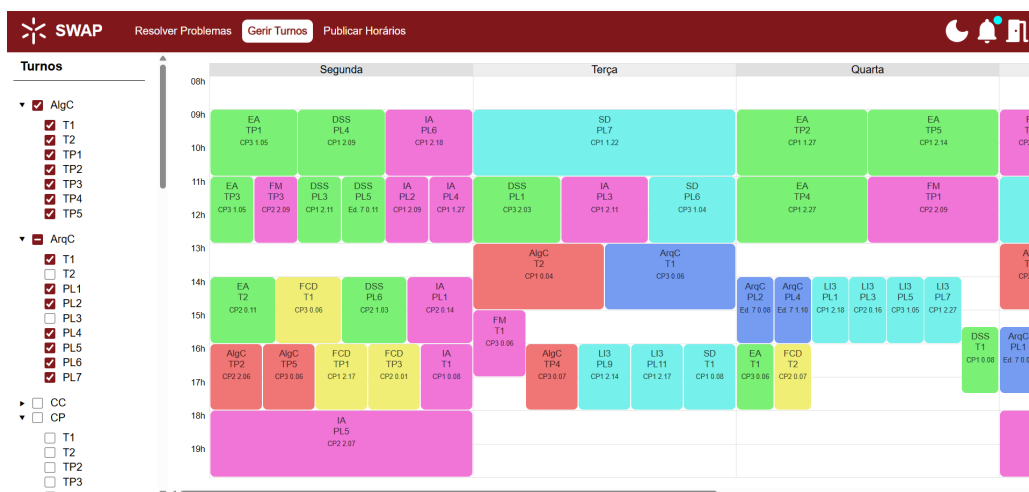


Figura 27: Página “Gerir Turnos” com alguns turnos por selecionar.

Clicando num turno, o utilizador será redirecionado para a página “Gerir Turno”, onde, para além de outras funcionalidades descritas posteriormente, pode consultar a sua lotação:

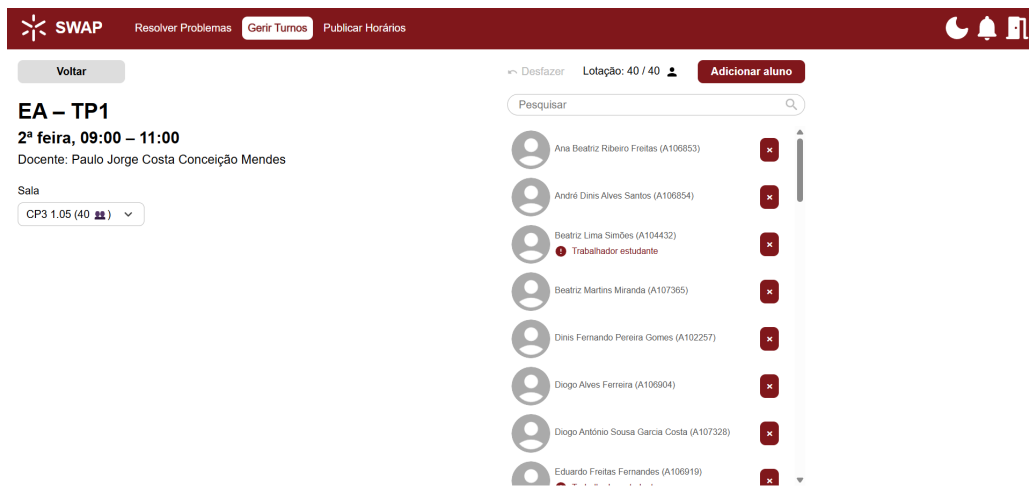


Figura 28: Página “Gerir Turno”.

### - Consultar alunos não colocados:

Ao entrar na aplicação, a página “Resolver Problemas” é apresentada ao diretor de curso, e mostra os alunos que não estão colocados num turno e os pedidos de troca de turno. Note-se que esta página também é acessível a partir da barra de navegação. A lista de problemas é pesquisável, tornando mais fácil encontrar um problema específico em listas de grande dimensão.

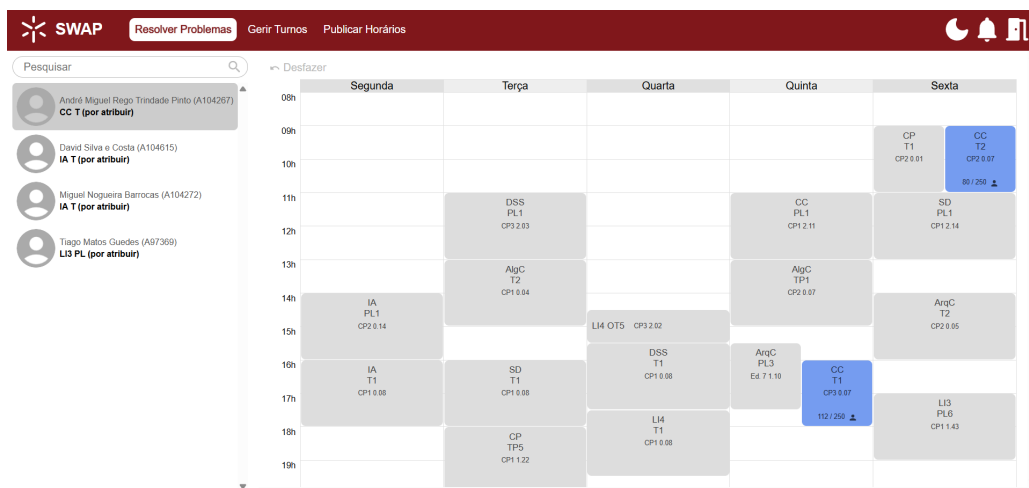


Figura 29: Página “Resolver Problemas”.

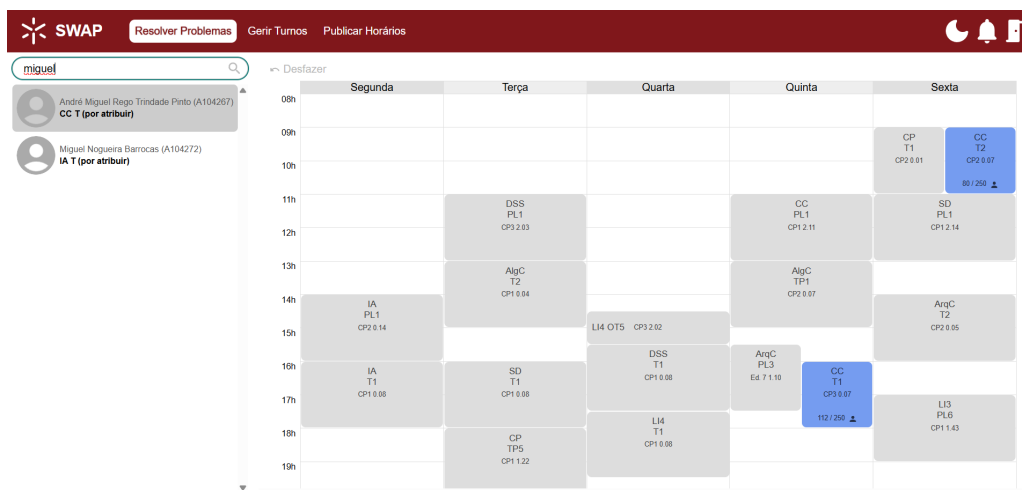


Figura 30: Página “Resolver Problemas” durante uma pesquisa por um aluno.

### - Alocação manual:

O diretor de curso pode, nesta página, clicar num aluno da lista para o tentar alocar a um turno. Pode ver o horário desse aluno e os turnos seleccionáveis, bem como a lotação dos mesmos, para assim tentar escolher um turno com vagas disponíveis que não crie conflitos no horário. Para colocar o aluno nesse turno de forma provisória, basta clicar no turno pretendido.

### - Publicar horário:

Para fazer com que as mudanças efetuadas fiquem guardadas e afetem o horário dos alunos, o diretor de curso deve aceder à página “Publicar Horários” e clicar no botão com o mesmo nome. Esta página apresenta sempre os problemas por revolver, lembrando ao diretor de curso de que pode estar a cometer um erro caso esteja a publicar horários ainda com turnos por atribuir e pedidos por satisfazer.

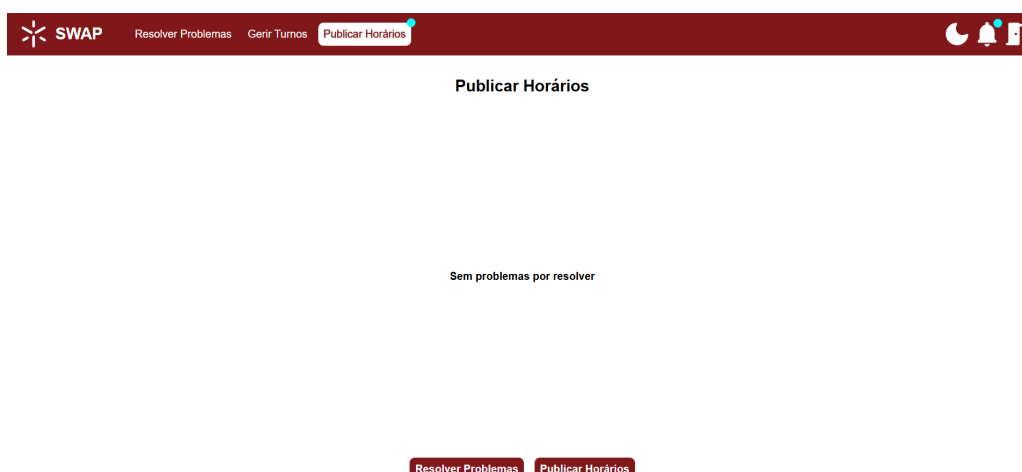


Figura 31: Página “Publicar Horários”.

Após publicar os horários, o utilizador será redirecionado para a página “Resolver Problemas”, e surgirá uma confirmação de que os horários foram publicados:

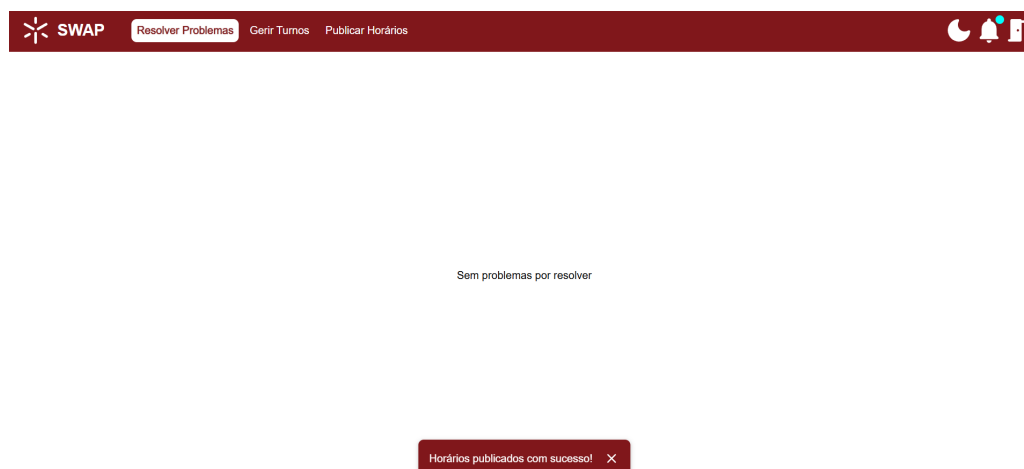


Figura 32: Confirmação da publicação de horários.

### 6.1.2 Cenário 2

Para as operações descritas no cenário 2, em que o diretor de curso responde a um pedido de mudança de sala, ele terá de realizar as seguintes ações:

#### - Ver os pedidos:

Para ver os pedidos feitos pelos docentes, o diretor de curso irá consultar as suas notificações, clicando no botão com ícone de sino, no canto superior direito da página. Sobrevoando o cursor ao pedido, terá três opções: rejeitá-lo, aceitá-lo ou visualizá-lo:

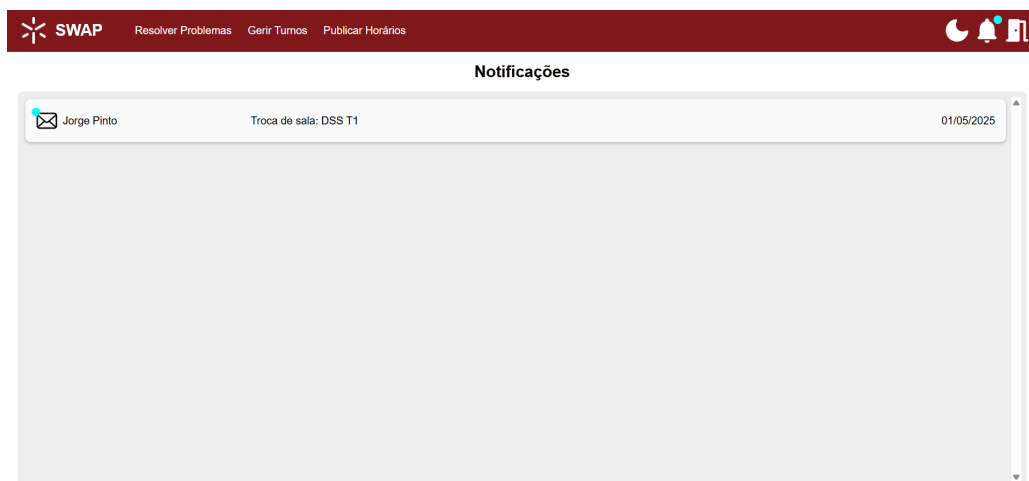


Figura 33: Notificação de troca de sala na página “Notificações do Diretor de Curso”.

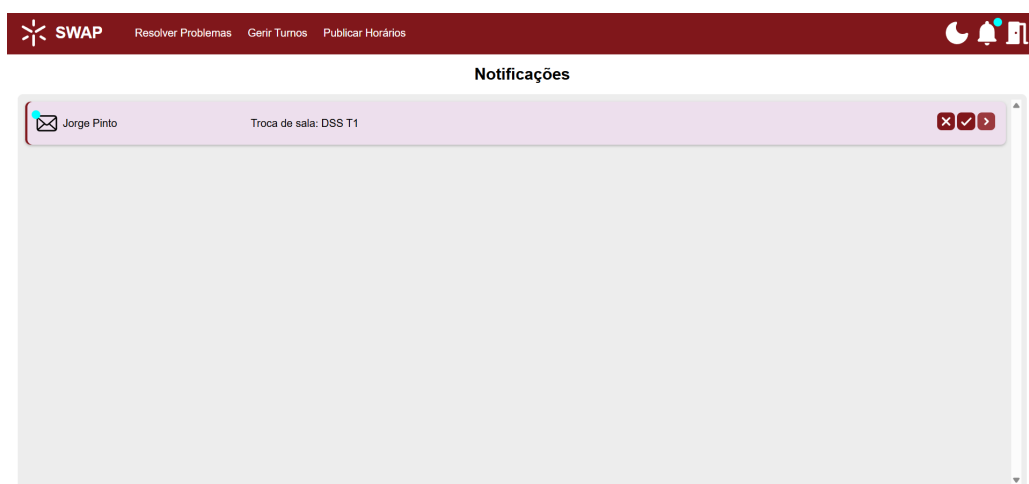


Figura 34: Opções de uma notificação na página “Notificações do Diretor de Curso”.

As opções de rejeitar e aceitar dão o pedido como resolvido, mostrando ao seu remetente se ele foi aceite ou rejeitado. A opção de visualizar o pedido leva o utilizador para a página onde este pode ser resolvido. Neste caso, o diretor, clicando nessa última opção, é redirecionado para a página “Gerir Turno” (figura 28).

#### - Consultar as salas disponíveis:

Nesta página, clicando no *dropdown* “Sala”, o diretor pode ver as salas disponíveis no horário daquele turno e a capacidade de cada sala. Neste exemplo, pode ver-se que a sala CP2 0.05 tem uma maior capacidade do que a sala atual, pelo que o diretor pode escolher mudar o turno para essa sala, selecionando-a no *dropdown*.

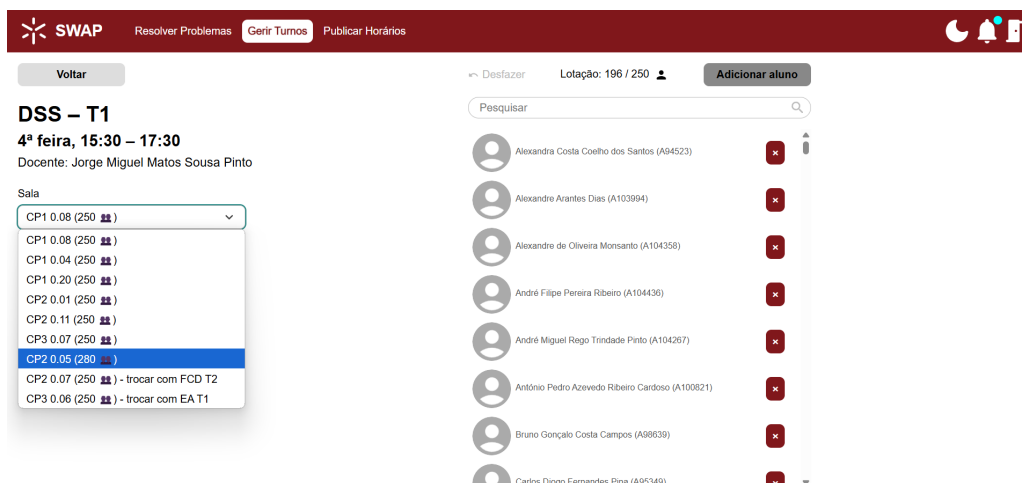


Figura 35: Seleção da sala de um turno na página “Gerir Turno”.

### - Fazer a alteração da sala:

Para efetivar a alteração da sala no horário dos alunos, tal como no cenário 1, o diretor deve publicar o novo horário (figura 31).

### 6.1.3 Cenário 3

Neste cenário, o diretor de curso vê que existe um pedido de troca de turno, e tenta atender a esse pedido. Para o fazer, realiza as seguintes ações na aplicação:

### - Verificar que recebeu um pedido:

Como mencionado anteriormente, ao entrar na aplicação, a página “Resolver Problemas” é apresentada ao diretor de curso. É nesta página que se encontram também os pedidos efetuados pelos alunos.

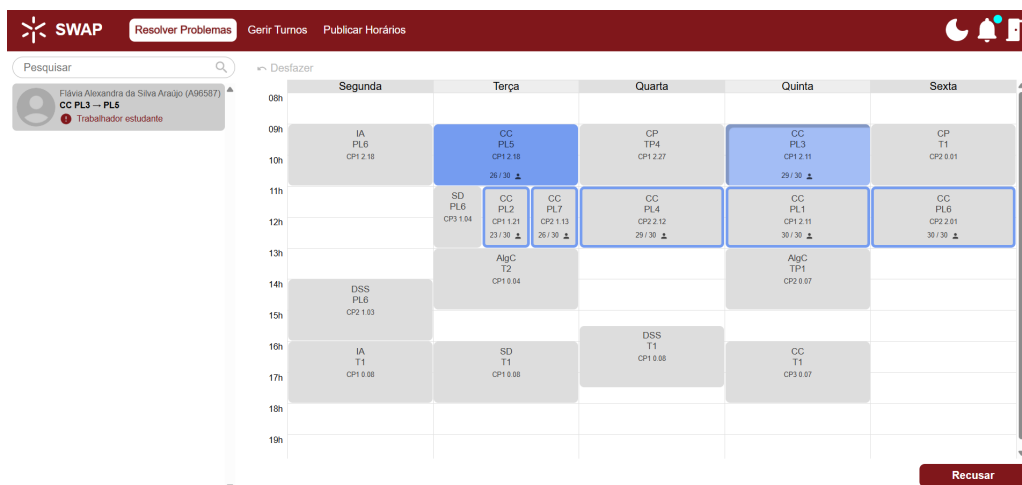


Figura 36: Página “Resolver Problemas” com um pedido de troca de turno.

O pedido pode também ser visto na página “Notificações do Diretor de Curso”. Neste caso, o botão para visualizar um destes pedidos redireciona o utilizador para a página “Resolver Problemas”, e não “Gerir Turnos”.

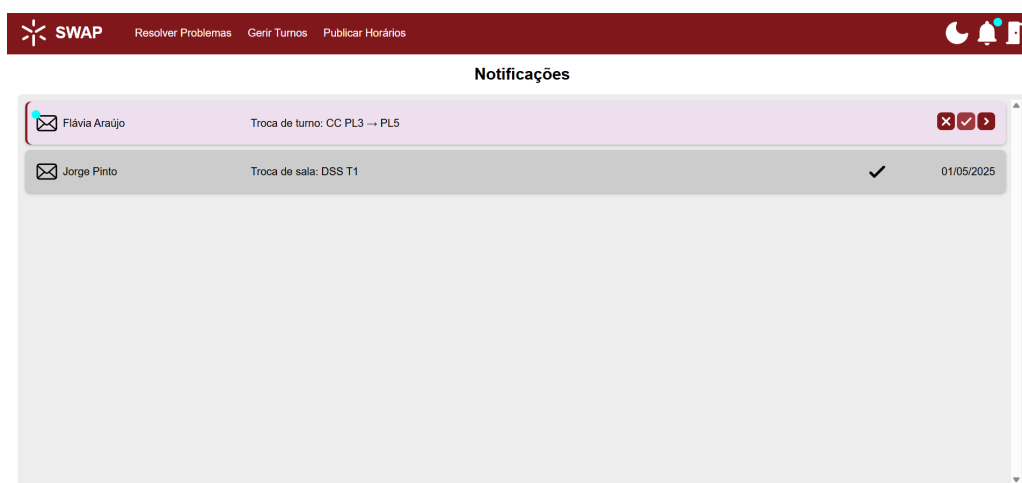


Figura 37: Página “Notificações do Diretor de Curso” com um pedido de troca de turno.

Nesta página, o diretor pode dar o pedido como aceite, sinalizando ao aluno que o pretende resolver, ou como rejeitado, sinalizando o contrário. Rejeitar um pedido também faz com que o mesmo não seja apresentado na página “Resolver Problemas” (figura 36).

#### - Verificar a capacidade do turno:

Na página “Resolver Problemas” (figura 36), é possível ver, no horário do aluno que efetuou o pedido, o turno para onde este se pretende mudar, os outros turnos da Unidade Curricular, e



a capacidade de cada um. Assim, pode verificar-se que o turno pretendido pelo aluno, o PL5, tem vagas disponíveis. No caso de não existir capacidade nesse turno, seria também visível se outros turnos à mesma hora têm capacidade e, no caso de não existam turnos a essa hora com vagas disponíveis, é também fácil de ver, pelo aviso na lista de pedidos, que o aluno em questão é trabalhador estudante, e que se pode abrir uma exceção e violar as restrições de capacidade dos turnos. Se for esse o caso, ao tentar alocar o aluno, irá aparecer uma mensagem de aviso, que informa que o turno pretendido está cheio, dando as opções de cancelar a operação, mudar a sala do turno (desativada no caso dos turnos práticos, porque o que limita a sua capacidade não é a sala), ou atribuir o turno mesmo assim.

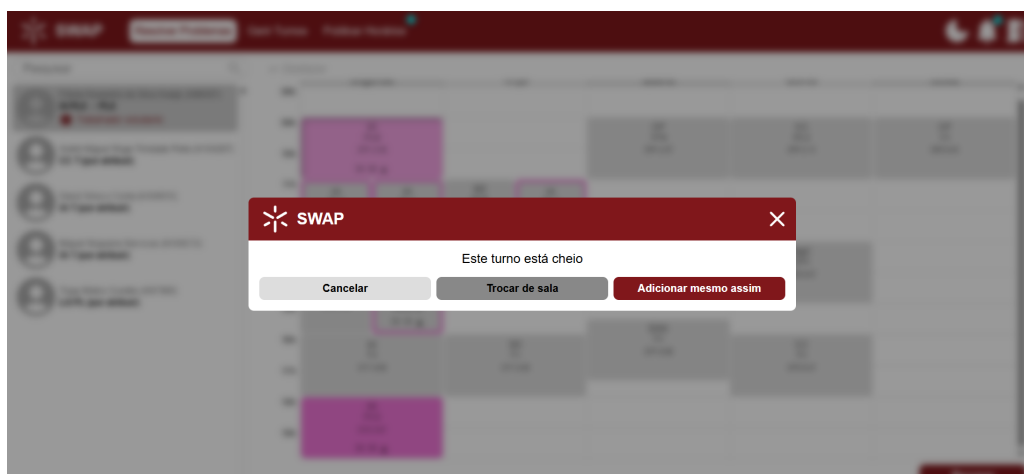


Figura 38: Popup que avisa que o turno ao qual se pretende adicionar um aluno está cheio.

#### - Realizar a troca de turno:

Clicando então no turno pretendido, o aluno fica inscrito no mesmo de maneira provisória. Para fazer com que esta mudança tenha efeito no horário do aluno, é necessário ir à página “Publicar Horários” e atualizar o horário dos alunos (figura 31). Logicamente, o diretor de curso pode e deve atender a vários pedidos dos alunos e apenas publicar os horários depois de concluir todas as alterações aos horários que desejava fazer.

### 6.1.4 Outras funcionalidades

#### - Gerir um turno:

O diretor de curso pode também, na página de gestão de um turno (figura 28), para além de consultar a lotação e trocar a sala de um turno, consultar e fazer a gestão dos alunos inscritos no mesmo. Na lista de alunos do lado direito, é possível pesquisar por nome do aluno, retirar

um aluno do turno clicando no X, e adicionar um aluno no botão do canto superior direito.

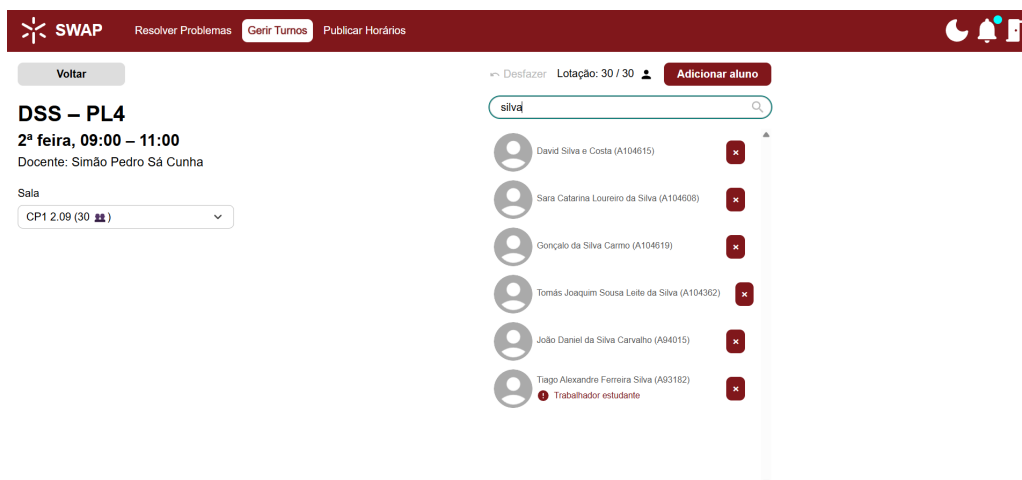


Figura 39: Pesquisa por alunos na página “Gerir Turno”.

No diálogo para adicionar um novo aluno, é apresentada uma lista dos alunos inscritos na Unidade Curricular a que o turno que se está a gerir pertence, com exceção dos alunos já inscritos no turno a ser gerido. Também é possível procurar alunos nesta lista. Para adicionar um aluno desta lista ao turno, clica-se no botão ‘+’ a si associado.

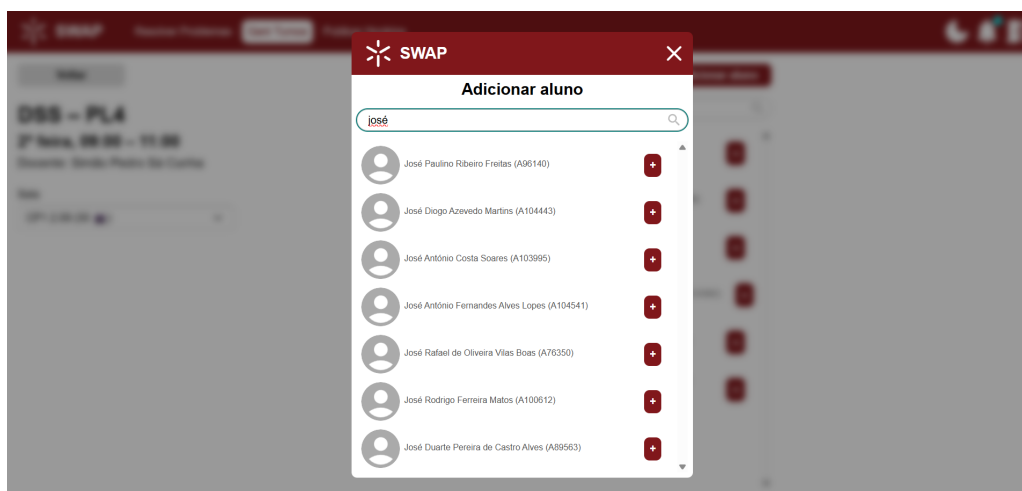


Figura 40: Diálogo para adicionar um aluno na página “Gerir Turno”.

Para estas mudanças terem efeito, também é necessário publicar o horário, como descrito anteriormente (figura 31).

- Mudar para o modo escuro:

Clicando no ícone de Lua no canto superior direito, o diretor de curso pode mudar a aplicação para o modo escuro, que passa a ter um aspeto como o que se apresenta abaixo:

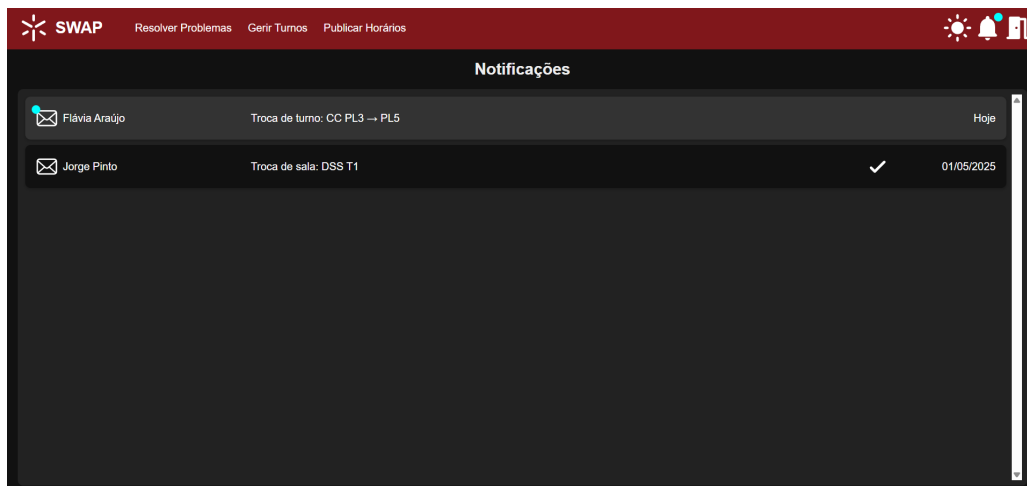


Figura 41: Exemplo de uma página no modo escuro.

Para voltar ao modo claro, é necessário clicar no botão de novo, que agora tem um ícone de Sol.

#### - Terminar sessão:

Para terminar a sua sessão, o diretor de curso deve clicar no botão com um ícone de porta no canto superior direito da barra de navegação, voltando assim à página “Iniciar Sessão”.

## 6.2 Aluno

### 6.2.1 Cenário 4

Para um aluno realizar as operações descritas no Cenário 4, ou seja, entrar na aplicação, consultar o seu horário e enviar um pedido de troca de turno, os passos a efetuar são os seguintes:

#### - Iniciar sessão com as suas credenciais:

Para se autenticar, o aluno segue o mesmo processo que o diretor de curso na página “Iniciar Sessão” (figura 24).

## - Consultar o seu horário:

Quando o aluno inicia a sua sessão, a primeira página que vê é a página do seu horário:

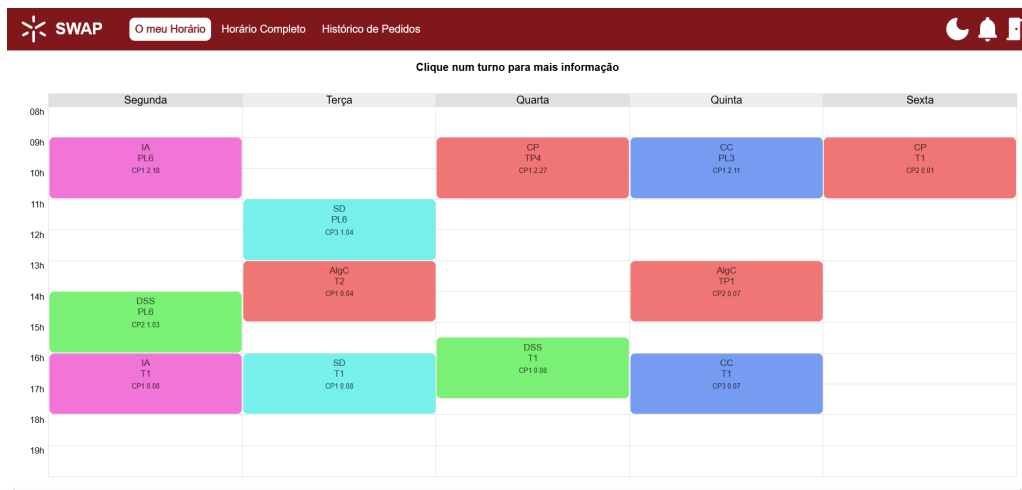


Figura 42: Página “O Meu Horário”.

## - Procurar uma troca de turno:

Supondo que, como descrito pelo cenário, o aluno pretende trocar os seus turnos de quinta-feira de manhã para outros dias, o primeiro passo será clicar nos turnos em questão, neste caso, o CC - PL3, que fará surgir o diálogo abaixo:

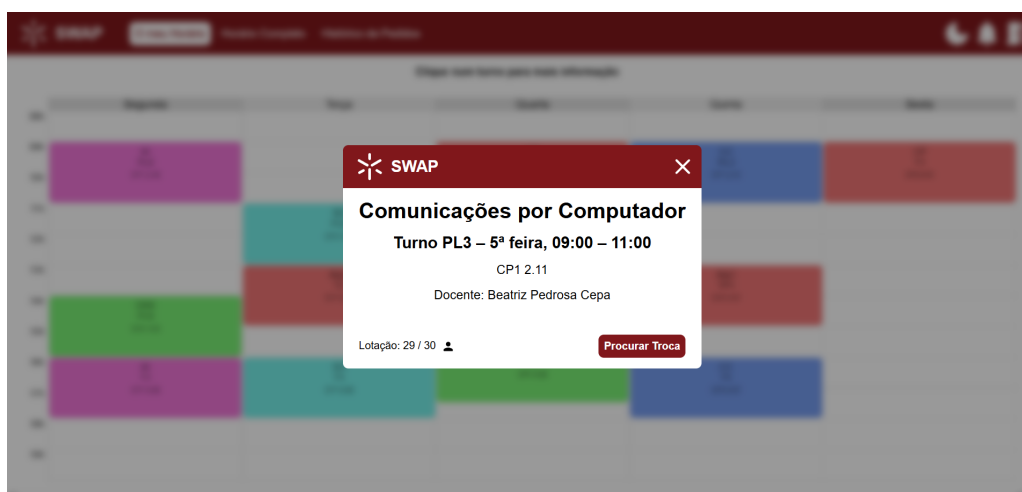


Figura 43: Diálogo com informações de um turno.

De seguida, o aluno clica no botão “Procurar Troca”, que o leva para o seguinte ecrã:

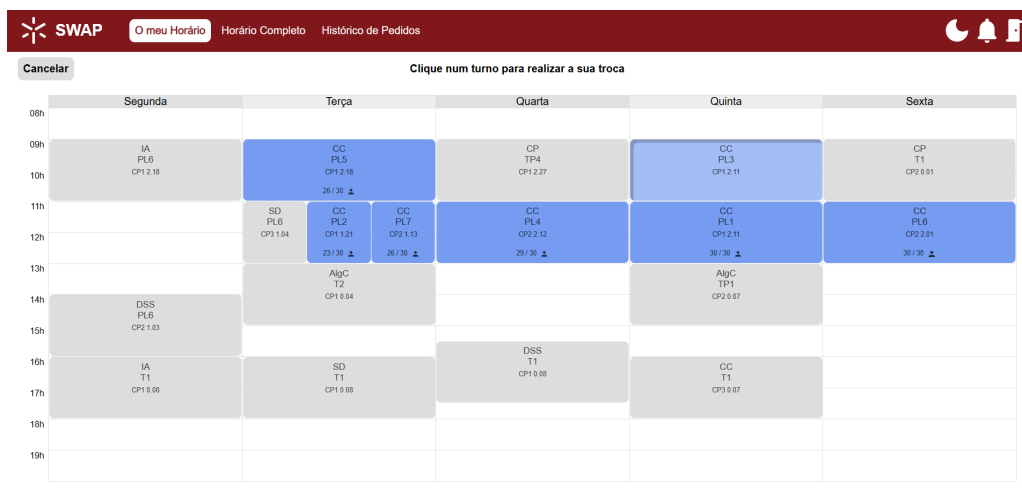


Figura 44: Troca de turno na página “O meu Horário”.

Aqui, o aluno pode ver o turno que pretende trocar, as opções de turnos para onde pode ir, as suas capacidades, e o seu horário atual, de forma a poder ver com facilidade se a troca que deseja fazer causará ou não conflitos no seu horário. Por exemplo, caso o aluno selecione trocar para o turno CC - PL5, deparar-se-á com o diálogo abaixo:

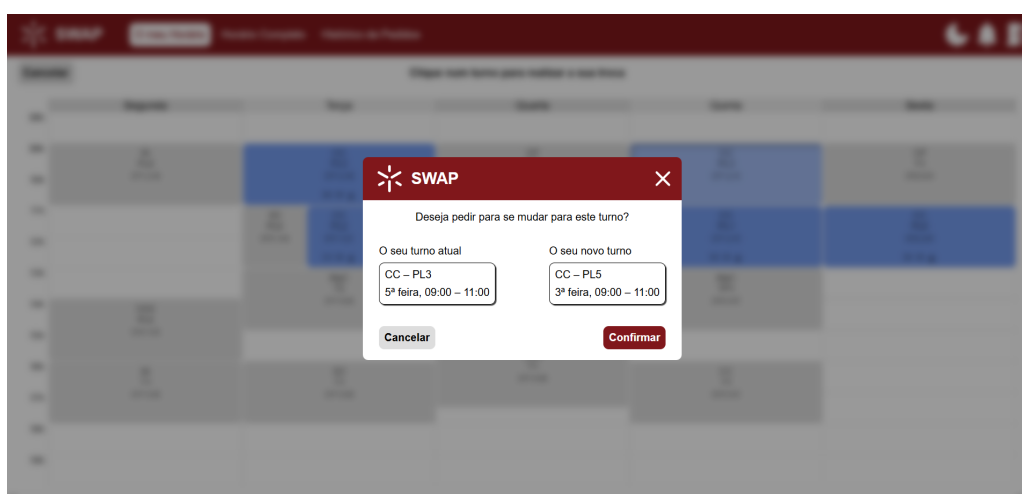


Figura 45: Pedido de confirmação de troca de turno.

Clicando em “Confirmar”, o pedido será enviado ao diretor de curso, e a seguinte mensagem de confirmação aparecerá por cima do horário do aluno:



Figura 46: Confirmação de sucesso na submissão de um pedido de troca de turno.

## 6.2.2 Outras funcionalidades

O aluno, para além do descrito acima, tem também as seguintes opções:

### - Ver o horário completo:

Se o aluno, através da barra de navegação, for para a página “Horário Completo”, pode ver o horário de todos os turnos das Unidades Curriculares em que está inscrito, com destaque para os turnos do seu horário. Para organizar esta grande quantidade de informação, o aluno pode, na barra lateral, seleccionar os turnos que pretende que lhe sejam apresentados:

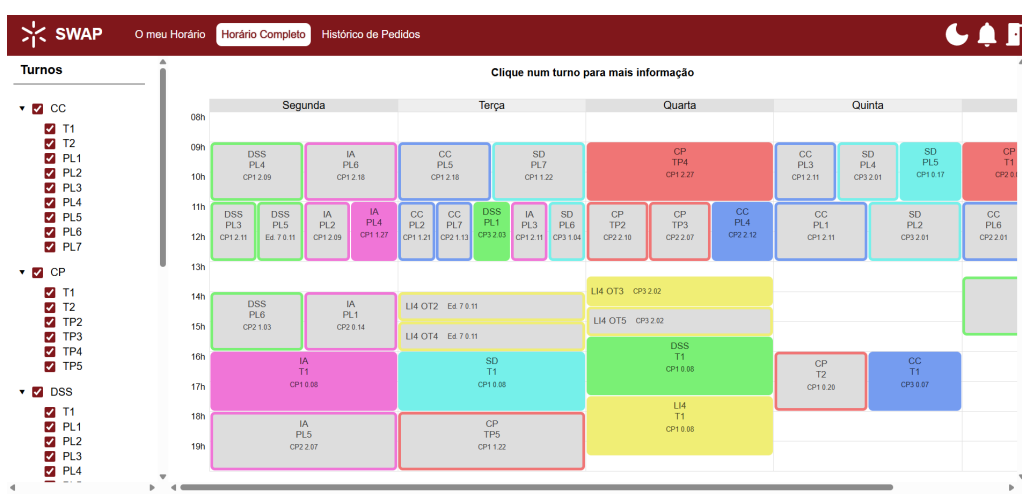


Figura 47: Página “Horário completo”.

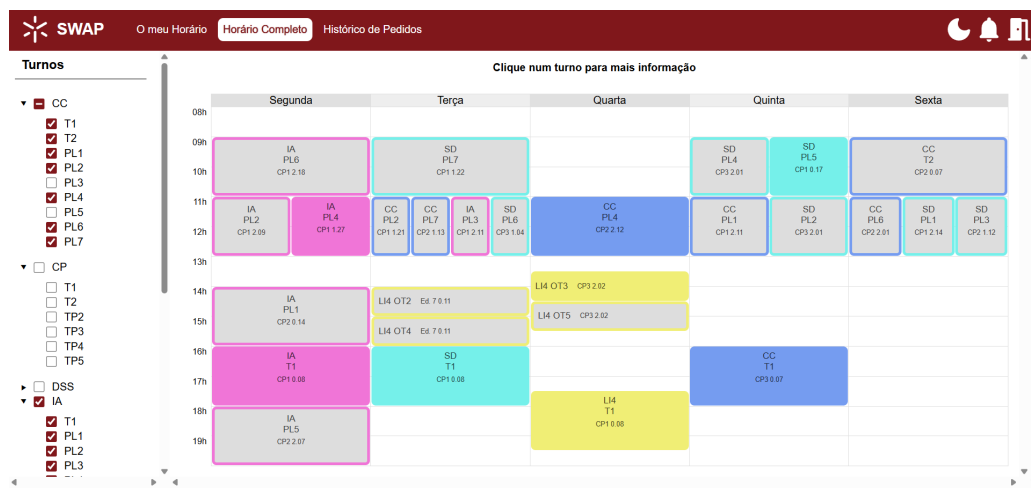


Figura 48: Página “Horário completo” com alguns turnos por seleccionar.

Esta página pode ser útil para alunos que desejam fazer mudanças ao seu horário mais complexas do que a troca de um ou dois turnos. Deste modo, podem ver mais informação, e seleccionar exactamente a informação que desejam ver (o seu horário atual, os turnos que desejam, ...).

Nesta página também é possível clicar num turno para ver mais informações sobre o mesmo e procurar uma troca, tal como na página “O meu Horário”. Durante uma troca de turnos, diferentes turnos são estilizados de diferente modo para realçar as opções de troca possíveis, o horário atual do aluno, outros turnos visíveis, *etc.*

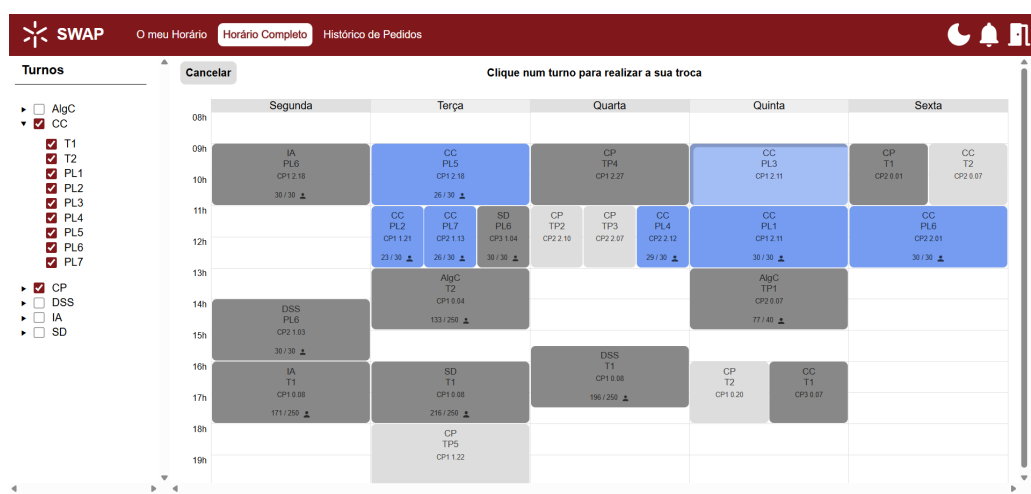


Figura 49: Procura de turno na página “Horário Completo”.

## - Ver o Histórico de Pedidos:

O aluno tem também a possibilidade de consultar os pedidos de troca de turno que já realizou e

o seu estado (pendente, aceite ou rejeitado), navegando para a página “Histórico de Pedidos”:

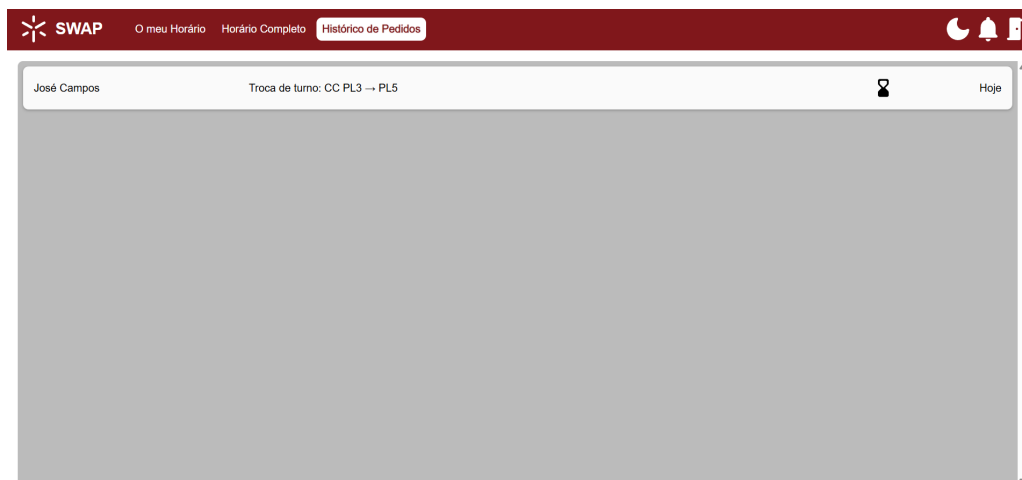


Figura 50: Página “Histórico de Pedidos” com pedido pendente.

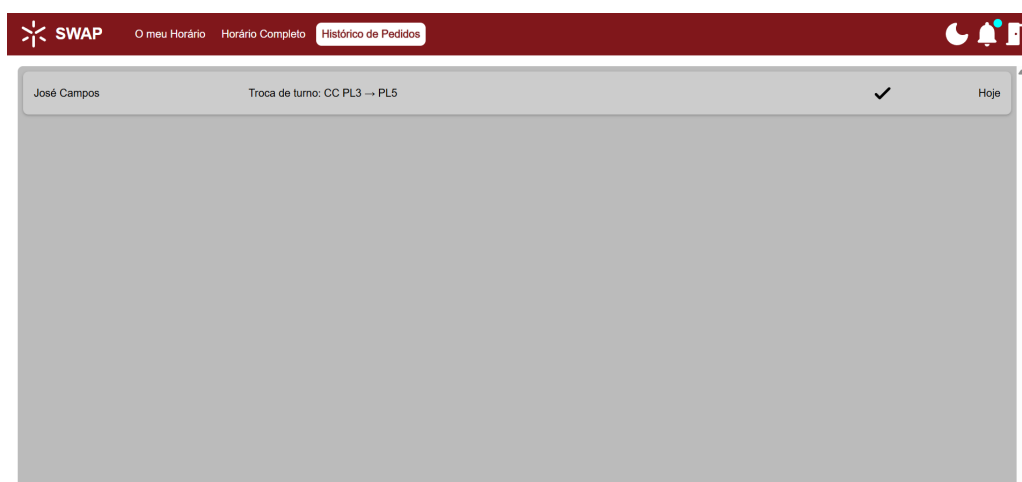


Figura 51: Página “Histórico de Pedidos” com pedido aceite.

#### - Ver as notificações:

O aluno pode também ver as suas notificações, que correspondem a mudanças no seu horário, clicando no botão com um ícone de sino no canto superior direito:



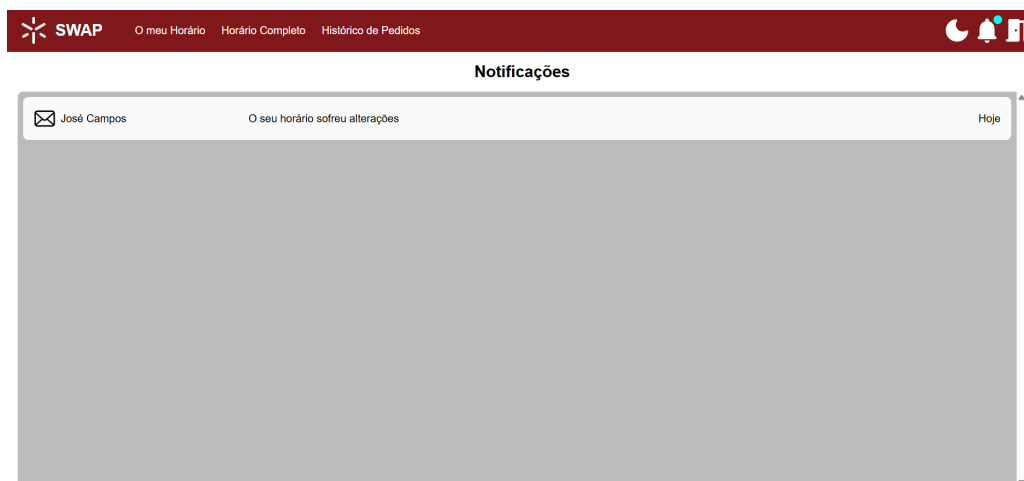


Figura 52: Página “Notificações do aluno”.

#### - Mudar para o modo escuro:

Tal como o diretor de curso, o aluno pode também mudar a aplicação para o modo escuro (figura 41).

#### Terminar sessão:

Tal como o diretor de curso, o aluno termina sessão clicando no botão com um ícone de porta no canto superior direito da barra de navegação, voltando assim à página “Iniciar Sessão”.

## 7 Acessibilidade

Durante o processo de desenvolvimento da aplicação, foi feito algum esforço para a tornar acessível.

Em primeiro lugar, o código da aplicação tira proveito da semântica de várias *tags* HTML (`nav`, `li`, `button`, ...), pelo que se espera que leitores de ecrã sejam capazes de compreender as páginas sem recurso a ARIA *labels*, e oferecer uma boa experiência de utilização a utilizadores com deficiências visuais. Para validar esta e outras hipóteses, a extensão WAVE [11] e o programa Pa1ly [12] foram utilizados para procurar diversos problemas de acessibilidade, que o nosso grupo de trabalho procurou corrigir. Também no âmbito da percetibilidade, garantiu-se que os rácios de contraste entre o fundo e texto dos componentes eram aceitáveis, pelo menos no tema claro. No tema escuro, isto não se verifica, visto que a cor da Universidade do Minho, utilizada em vários elementos da interface, não contrasta bem com tons escuros. Mesmo assim,

o utilizador tem a opção de utilizar o tema claro caso o necessite.

No âmbito da operabilidade, o nosso principal foco do foi o de garantir que era possível navegar pelas várias páginas com o teclado de uma forma que faça sentido. A título de exemplo, é possível fechar diálogos e *toasts* com a tecla **Esc**, e é possível desfazer ações com **Ctrl+Z**. Ademais, foi preciso especial cuidado nos horários para estes serem facilmente navegáveis com a tecla **Tab**: como estes têm um *layout* tabular, linha a linha, foi preciso reajustar os valores de `tabindex` para garantir que os turnos eram visitados em ordem cronológica. Também se garantiu que os elementos seleccionáveis tinham a sua classe CSS `:focus-visible` estilizada.

No entanto, o principal problema que a aplicação sofre é a falta de responsividade perante diferentes tamanho de ecrã: são poucas as páginas e componentes que funcionam em ecrãs pequenos e verticais como os dos *smartphones*. Fazer as alterações necessárias ao *layout* das páginas para suportar variadíssimos tamanhos de ecrãs já envolveria bastante trabalho, mas também aumentaria significativamente o número de testes que se teriam de fazer a cada página: cada página teria de ser testada em vários navegadores e em vários tamanhos de ecrã, sendo que qualquer mudança ao *layout* envolveria refazer toda esta testagem. Devido ao tempo que isto consumiria, optámos por dedicar mais tempo a melhorar a *user experience* na aplicação, do que a a otimizar para diversos tamanhos de ecrã. No entanto, no futuro, se esta aplicação fosse usada em produção, *layout* responsivo seria um das primeiras funcionalidades a ser implementada.

## 8 Reflexão sobre a Aplicação

Devido a falta de tempo, não foi possível avaliar a aplicação desenvolvida com utilizadores reais. Mesmo assim, apresentam-se, nesta secção, alguns comentários pessoais do nosso grupo sobre a aplicação final desenvolvida.

Em primeiro lugar, é necessário notar que a aplicação cumpre os requisitos propostos, permitindo ao diretor de curso e aos alunos executar as ações nos cenários do enunciado do trabalho prático. Ademais, é necessário considerar que o modelo Figma da interface previamente construído foi avaliado de forma positiva, tanto heurísticamente como pela equipa docente. Visto que a aplicação implementada seguiu este modelo do modo mais próximo possível, acredita-se que a aplicação também providenciará uma boa experiência para o utilizador.

No entanto, surgiram alguns aspetos que podiam ser melhorados numa eventual próxima iteração no desenvolvimento da aplicação. Em primeiro lugar, em páginas que dão a possibilidade ao utilizador de desfazer uma ação, também é importante dar a possibilidade de refazer uma ação previamente desfeita, algo que está em falta na aplicação desenvolvida. Ade-

mais, na forma como as notificações são geridas, a experiência do utilizador melhoraria se as notificações fossem automaticamente aceites conforme as ações do utilizador. Por exemplo, se o diretor de curso visse o conteúdo de um pedido de mudança de sala e alterasse a sala em que o turno é lecionado, a notificação deveria automaticamente ser marcada como aceite. Estes pequenos detalhes são cruciais para assegurar uma boa experiência de utilização e, apesar de poucos, acreditamos que há espaço para melhoria na aplicação desenvolvida.

## 9 Conclusão

Em suma, ao longo deste semestre, foi levado a cabo o processo completo de desenvolvimento de uma interface, partindo da sua modelação, passando pela sua implementação, e terminando na sua testagem e avaliação. Em relação à segunda fase do trabalho prático, devido ao grande número de ferramentas utilizado, foi possível aprender um pouco sobre diversos aspetos de desenvolvimento Web, apesar de não ter sido possível aprofundar muito o nosso conhecimento em áreas específicas. Por muito que não tenha sido possível realizar a fase da avaliação da aplicação com o detalhe que era desejado, acreditamos que o produto desenvolvido não só cumpre os requisitos propostos, como também o faz apresentando uma boa experiência de utilização.

## 10 Bibliografia

- [1] “The Progressive JavaScript Framework”. Figma. Accessed: Apr. 30, 2025. [Online.] Available: <https://vuejs.org/>
- [2] “Figma: Collaborative Interface Design Tool”. Figma. Accessed: Mar. 13, 2025. [Online.] Available: <https://www.figma.com/>
- [3] “TypeScript”. TypeScript. Accessed: Apr. 30, 2025. [Online.] Available: <https://www.typescriptlang.org/>
- [4] “Launch your idea to the web fast with Blazor”. Blazor. Accessed: Apr. 30, 2025. [Online.] Available: <https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps/blazor>
- [5] “Pinia: The intuitive store for Vue.js”. Pinia. Accessed: Apr. 30, 2025. [Online.] Available: <https://pinia.vuejs.org/>
- [6] “Pinia Plugin Persistedstate: Configurable persistence of Pinia stores”. Pinia Plugin Persistedstate. Accessed: Apr. 30, 2025. [Online.] Available: <https://prazdevs.github.io/pinia-plugin-persistedstate/>

- [7] “json-server”. GitHub. Accessed: Apr. 24, 2025. [Online.] Available: <https://github.com/typicode/json-server>
- [8] “NPM”. NPM. Accessed: Apr. 24, 2025. [Online.] Available: <https://www.npmjs.com/>
- [9] “ESLint”. ESLint. Apr. 24, 2025. [Online.] <https://eslint.org/>
- [10] “Prettier”. Prettier. Apr. 24, 2025. [Online.] <https://prettier.io/>
- [11] “WAVE - Web Accessibility Evaluation Tool”. WAVE. Accessed: May 4, 2025. [Online.] Available: <https://wave.webaim.org/extension/>
- [12] “Pa11y”. Pa11y. Accessed: May 4, 2025. [Online.] Available: <https://pa11y.org/>

## 11 Anexo – Enunciado do Trabalho

### Interface Pessoa-Máquina

Licenciatura em Engenharia Informática

Departamento de Informática

Universidade do Minho

---

Enunciado do Trabalho

---

2024/2025

#### Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Objetivo do trabalho</b>	<b>1</b>
<b>3</b>	<b>O domínio</b>	<b>1</b>
<b>4</b>	<b>Cenários de Utilização</b>	<b>2</b>
<b>5</b>	<b>Utilizadores</b>	<b>3</b>
<b>6</b>	<b>Realização do trabalho</b>	<b>4</b>
6.1	Proposta de interface . . . . .	4
6.2	Entrega final . . . . .	5
<b>7</b>	<b>Avaliação</b>	<b>5</b>
<b>8</b>	<b>Grupos de Trabalho</b>	<b>5</b>

## 1 Introdução

Este documento apresenta o enunciado do trabalho prático da Unidade Curricular (UC) de Interface Pessoa-Máquina (IPM) para o ano letivo 2024/2025. **Leia-o com atenção.** Quaisquer dúvidas devem ser esclarecidas junto dos docentes da UC.

## 2 Objetivo do trabalho

Em cursos de grande dimensão, como é o caso da Licenciatura em Engenharia Informática (LEI), os alunos têm de ser distribuídos por turnos para viabilizar a realização das aulas. A gestão dessa distribuição por turnos é um processo complexo, uma vez que é necessário compatibilizar os turnos das diferentes Unidades Curriculares (UC) de cada ano do curso (e, eventualmente, de diferentes anos do curso), de modo a que os alunos não tenham colisões no seu horário.

Historicamente, os docentes de cada UC geriam a definição dos turnos da sua UC. Esta solução mostrou não ser escalável, criando situações em que os alunos tinham dificuldades em obter horários sem colisões e atrasando o início das aulas.

Uma forma de facilitar o processo é centralizar a criação dos turnos nas direções de curso. No entanto, essa criação é complexa e demorada, consumindo recursos, pelo que só é viável se as listas de inscritos forem conhecidas com antecedência suficiente, o que nem sempre acontece (por exemplo, no primeiro semestre do ano). Além disso, tem as suas próprias desvantagens, uma vez que não permite aos alunos controlo sobre o seu horário.

**Neste projeto é-lhe pedido que conceba e implemente a interface de utilizador de um sistema para auxiliar o diretor de curso na gestão dos turnos práticos de um semestre de um curso.** Para efeitos deste trabalho, assume-se que são conhecidas as UC de cada ano do curso nesse semestre, o horário do curso e as listas de inscritos a cada UC, bem como uma primeira distribuição dos alunos por turnos.

## 3 O domínio

Para a realização deste trabalho recorra aos conhecimentos que tem sobre cursos, horários, UC e alunos. Considere que um curso é organizado em anos, tendo cada ano dois semestres<sup>1</sup>. Cada semestre do curso tem um conjunto de UC (algumas

---

<sup>1</sup>Note que é também possível pensar nos cursos em termos de semestres: semestre um, dois, três, quatro, etc. — ou seja, primeiro semestre do primeiro ano, segundo semestre do primeiro ano, primeiro semestre do segundo ano, segundo semestre do segundo ano, etc.

poderão ser UC opcionais, com várias alternativas), podendo os alunos estar inscritos a UC de anos diferentes.

Ao pensar na sua solução para a interface do sistema tenha em atenção que diferentes tipos de aula podem ter limites de alunos por turma diferentes. No caso das aulas teóricas, o limite é a capacidade da sala a que a aula está alocada no horário. No caso de aulas teórico-práticas (TP) e práticas laboratoriais (PL) o limite é definido como o menor valor entre o tamanho das turmas, definido UC a UC, e a capacidade da sala atribuída.

## 4 Cenários de Utilização

Por forma a facilitar o levantamento de requisitos, apresentam-se aqui um conjunto de cenários de utilização da aplicação. O sistema que vão desenvolver deverá ser capaz de os suportar. No entanto, poderão incluir na vossa análise de requisitos a vossa experiência com a inscrição e frequência em turnos práticos de UC do curso.

### Cenário 1

O diretor de curso acede à aplicação, usando o seu identificador de funcionário e palavra-passe, para consultar o resultado da geração dos turnos pela aplicação. Consulta os horários de cada UC, para avaliar a percentagem de ocupação de cada turno. Anota nos seus apontamentos as UC com ocupações de turnos mais desequilibradas, para eventuais ajustes aos horários do curso.

Repara, então, que o algoritmo foi incapaz de colocar 45 alunos, por não conseguir respeitar todas as preferências, sem evitar conflitos nos seus horários. O diretor de curso procede então à alocação manual desses alunos aos turnos disponíveis. Em caso de conflito nos horários dos alunos, o sistema avisa-o da situação. Na impossibilidade de evitar alguns desses conflitos, o diretor de curso opta por dar prioridade aos alunos de primeira inscrição, fazendo a distribuição manual de modo a minimizar conflitos a esses alunos.

Após terminar a distribuição, o diretor de curso publica os horários dos alunos.

### Cenário 2

Durante a utilização da aplicação, o diretor de curso nota que recebeu (na aplicação) um pedido de um docente para que um turno de uma UC seja mudado para uma sala maior. Consulta o horário da UC e as salas disponíveis e verifica que existe uma sala com mais capacidade, disponível no horário do turno em causa<sup>2</sup>. Faz, en-

---

<sup>2</sup>A Direção de Curso tem um conjunto de salas atribuídas ao curso e faz a sua gestão.

tão, a alteração é enviada (através da aplicação) uma mensagem a confirmar a satisfação do pedido (caso não o tivesse podido satisfazer, seria enviada uma mensagem também, nesse caso a avisar da impossibilidade de satisfazer o pedido).

### **Cenário 3**

Durante a utilização da aplicação, o diretor de curso nota que recebeu um pedido da Maria para ser mudada de turno numa UC. Consulta os turnos da UC e verifica que existe capacidade disponível no turno pretendido. Faz a alteração e é enviada uma mensagem a confirmar a satisfação do pedido

Caso não existisse capacidade no turno pretendido, o diretor de curso iria verificar se existia outro turno com capacidade no mesmo horário e, se sim, mudaria a Maria para esse turno, avisando-a.

Caso não existisse nenhum turno com capacidade disponível no horário pretendido, mas a Maria fosse uma aluna com estatuto, o diretor de curso faria, ainda assim, a alteração (neste caso, o turno ficaria com mais alunos do que a sua capacidade e o diretor de curso iria, posteriormente, avaliar se conseguia mudar o turno para uma sala maior).

### **Cenário 4**

A Maria entra na aplicação com as suas credências e consulta o horário que lhe foi atribuído. Percebe que tem aulas na quinta de manhã e envia um pedido ao diretor de curso a pedir a mudança desses turnos para outros, em horários que lhe agradam mais e em que vê que existe capacidade disponível.

## **5 Utilizadores**

Para auxiliar o desenvolvimento da interface, foram desenvolvidas os seguintes dois perfis de utilizador:

### **Perfil 1**

**Nome:** José

**Idade:** 35 anos

**Profissão:** Docente do ensino superior; diretor de curso

**Localização:** Braga

**Educação:** Doutoramento em informática

**Interesses:** Automóveis, futebol, séries



**Objetivos:** Manter o curso a funcionar de forma eficiente, minimizando o esforço burocrático de gestão; maximizar as possibilidades de aprendizagem dos alunos.

**Desafios:** Sucesso do curso leva a um número crescente de alunos inscritos

**Soluções:** Automatizar processos para diminuir trabalho burocrático

**Citação:** "Eu gosto do que faço, mas a direção de curso ocupa-me muito tempo com tarefas de gestão com pouco valor acrescentado para o curso. O novo sistema deverá ajudar a minimizar isso. Espero que seja de utilização eficaz e eficiente."

## Perfil 2

**Nome:** Maria

**Idade:** 18 anos

**Profissão:** Estudante

**Localização:** Braga

**Educação:** Frequência do primeiro ano do curso

**Interesses:** Tecnologia, automobilismo, jogos de computador

**Objetivos:** Realizar o curso. Ter um horário de acordo com o que lhe convém.

**Desafios:** Adaptar-se e compreender o funcionamento do curso na universidade

**Soluções:** Perguntar aos colegas que conhece

**Citação:** "Ainda não percebi bem como funciona isto dos turnos, espero não ficar com um horário cheio de buracos. Disseram-me que era bom se não tivesse aulas na quinta de manhã."

## 6 Realização do trabalho

O trabalho será realizado em duas fases.

### 6.1 Proposta de interface

Prototipagem de uma proposta de interface, a concluir até **15 de março**. Para além do protótipo em Figma, deverá incluir uma breve análise em que se descreva de que forma as *guidelines* de usabilidade foram tidos em consideração (podem escolher qual dos conjuntos utilizar) e de que forma a interface proposta responde às necessidades dos dois perfis definidos.

O protótipo valerá 25% da nota final.

## 6.2 Entrega final

Implementação da interface proposta utilizando Vue.js. O resultado desta fase terá de se entregar até **2 de maio**. Após a definição dos grupos de trabalho, serão criados repositórios GitHub para apoiar a realização do trabalho. Oportunamente será fornecido um *backend* implementado com recurso a json-server.

A implementação vale 75% da nota final.

## 7 Avaliação

Para a discussão do trabalho deverão preparar uma apresentação com a duração máxima de 10 minutos. Esta apresentação deverá descrever a solução e a abordagem seguida para a atingir, desde a definição do protótipo e sua avaliação, até à implementação e demonstração da solução final. A apresentação deverá terminar com uma análise crítica dos resultados obtidos. Após essa apresentação, seguir-se-á um período de discussão do trabalho de até 20 minutos.

A apresentação e discussão final do trabalho será realizada nas semanas de 5 a 16 de maio, em horários a combinar. A **presença** na discussão do trabalho é **obrigatória**.

## 8 Grupos de Trabalho

Os grupos de trabalho deverão obrigatoriamente ser constituídos por de 3 a 5 elementos. O registo dos grupos de trabalho deve ser realizado no Blackboard, estando já disponível. A fase de **registo dos grupos termina a 28 de fevereiro às 17:30**.