



Departamento de Informática  
Escola de Engenharia  
Universidade do Minho

# Gestor de Turnos

**Interface Pessoa-Máquina**  
Licenciatura em Engenharia Informática  
2024/2025

[Link para o projeto](#)



## Grupo 01

Ana Cerqueira, a104188  
Diogo Barros, a100751  
Diogo Esteves, a104004  
José Lopes, a104541  
José Matos, a100612

04/05/2025

# **Conteúdo**

<b>1</b>	<b>Objetivos do Trabalho</b>	<b>2</b>
<b>2</b>	<b>Descrição de Alterações ao Protótipo</b>	<b>3</b>
<b>3</b>	<b>Descrição da Implementação</b>	<b>4</b>
3.1	Estrutura e Decomposição em Componentes . . . . .	4
3.2	Lógica de Negócio . . . . .	4
3.3	Estrutura da Base de Dados . . . . .	4
3.4	Bibliotecas e Ferramentas Utilizadas . . . . .	5
3.5	Comunicação com o Backend . . . . .	5
<b>4</b>	<b>Reflexão sobre a Aplicação Desenvolvida na Perspetiva do Utilizador</b>	<b>6</b>
4.1	Pontos Fortes . . . . .	6
4.2	Pontos Fracos . . . . .	6
4.3	Testes de Acessibilidade com Pa11y . . . . .	7
4.4	Considerações Finais . . . . .	7
<b>5</b>	<b>Manual de Utilização</b>	<b>8</b>
5.1	Descrição das Páginas . . . . .	8
5.2	Descrição dos Popups . . . . .	11
<b>6</b>	<b>Reflexão sobre o Trabalho</b>	<b>13</b>

## 1 Objetivos do Trabalho

O principal objetivo deste trabalho consistiu no desenvolvimento de uma aplicação web interativa que permita a gestão de turnos académicos por parte dos estudantes e do diretor do curso, com funcionalidades de troca rápida de turnos. A aplicação visa simular um sistema de comunicação assíncrona entre estudantes, facilitando o envio, receção e gestão de pedidos relacionados com os seus horários.

De forma mais concreta, pretendeu-se atingir os seguintes objetivos:

- Criar uma interface intuitiva onde o estudante possa consultar os seus turnos atribuídos.
- Permitir o envio de mensagens entre o diretor de curso e professores/estudantes.
- Gerir o estado de mensagens (enviadas, recebidas, lixo), criando uma caixa de correio digital.
- Permitir o diretor do curso de gerir os turnos de cada uma das Unidades Curriculares.
- Permitir o diretor de curso consultar e gerir as alocações de cada um dos alunos, bem como consultar o perfil destes.

Além disso, foi de especial importância desenvolver um horário modular e responsivo, de forma a apresentar o máximo número de turnos ao mesmo tempo sem sacrificar a interface visual ou a experiência do utilizador.

## 2 Descrição de Alterações ao Protótipo

Durante o desenvolvimento da aplicação, algumas alterações ao protótipo inicialmente proposto na Fase 1 foram introduzidas, com o objetivo de melhorar a experiência do utilizador e simplificar a lógica de implementação. Estas alterações, justificadas tanto por questões de design como por motivos técnicos, são descritas de seguida:

- **Indicação visual de turnos cheios (*não implementada*):** estava inicialmente previsto que turnos com o número máximo de alunos fossem assinalados com uma cor distinta, acompanhados por um rótulo com o número atual e o número máximo de estudantes (ex: 35/40). No entanto, optou-se por não incluir esta funcionalidade na versão final, uma vez que, na prática, introduzia demasiado ruído visual na interface e dificultava a leitura dos turnos. A decisão foi tomada para preservar a clareza da interface.
- **Remoção da notificação após envio de mensagem:** no protótipo original, estava prevista uma notificação visual a confirmar o envio de mensagens. Esta funcionalidade foi retirada, passando o envio de mensagens a ocorrer de forma direta e silenciosa, com impacto imediato na caixa de correio. Esta decisão simplificou a experiência do utilizador e eliminou passos desnecessários, tornando o processo mais fluido.
- **Ícones de navegação sempre visíveis:** em vez de esconder os ícones de navegação baseado na página atual do utilizador, decidiu-se mantê-los sempre visíveis, mas com um aspeto atenuado (uma cor mais escura). Esta abordagem melhora a consistência da interface e ajuda o utilizador a perceber melhor o estado atual da aplicação, bem como os efeitos quando navega entre páginas.
- **Remoção do filtro por turno:** o filtro que permitia ao utilizador restringir a lista de turnos por turno foi removido. Esta decisão prendeu-se com a percepção de que o filtro, embora funcional, complicava a interface sem trazer valor significativo na maioria dos casos.

Estas alterações refletem uma abordagem centrada no utilizador, onde a clareza da interface e a simplicidade das interações foram priorizadas. Optou-se por sacrificar certas funcionalidades planeadas que, embora tecnicamente viáveis, não contribuíam positivamente para a experiência final.

## 3 Descrição da Implementação

A solução para a aplicação desenvolvida foi implementada com recurso ao framework Vue 3, utilizando a Composition API, e faz uso de diversas bibliotecas auxiliares para gestão de estado, comunicação com o backend e organização de componentes.

### 3.1 Estrutura e Decomposição em Componentes

A aplicação está dividida em múltiplas páginas (views), cada uma representando funcionalidades específicas. Os componentes foram organizados em três níveis principais:

- **Templates**: componentes de layout que organizam a estrutura visual da página (ex: `MessagesTemplate.vue`).
- **Views**: páginas que lidam com lógica de carregamento e ligação entre dados e os templates (ex: `MessagesPage.vue`).
- **Stores**: gestão de estado com Pinia, permitindo persistência local e sincronização com o backend (ex: `allocationStore.ts`, `messagesStore.ts`, `userInboxStore.ts`).

Além disso, as templates foram montadas a partir de componentes que, por sua vez, poderiam utilizar dentro de si outras componentes a níveis mais elementares. Por exemplo, a página de alunos (`StudentPage.vue`) monta a template (`StudentTemplate.vue`). Esta template é de seguida construída a partir de várias outras componentes elementares (como `ShiftSelector.vue`).

### 3.2 Lógica de Negócio

A lógica de negócio também foi separada por responsabilidades, onde cada uma das *stores* gere uma entidade da base de dados. Por exemplo:

- A `messagesStore` gere o conjunto global de mensagens e permite ações como aprovar, rejeitar, mover para o lixo e restaurar.
- A `userInboxStore` faz a gestão da caixa de entrada, enviados e lixo, por utilizador.
- A `allocationStore` gere as alocações dos estudantes a turnos específicos.
- A `scheduleStore` mantém os eventos e turnos disponíveis.

Deste modo, a aplicação segue um padrão reactivo, onde alterações feitas pelos utilizadores atualizam automaticamente o estado local e são depois propagadas para o backend a partir de operações de sincronização.

### 3.3 Estrutura da Base de Dados

A base de dados utilizada é um ficheiro JSON servido por `json-server`. A estrutura foi desenhada de forma a representar entidades fundamentais como mensagens, utilizadores, eventos e alocações. As principais entidades presentes são:

- **courses**: cada UC é representada por nome, abreviação, semestre e curso.

```
{  
    "id": 3,  
    "courseId": "Laboratórios de Informática I",  
    "name": "LI1",  
    "semester": 1,  
    "degreeId": 1  
}
```

- **shifts**: cada turno contém o nome, a sua UC, a sala, o professor, o número de alunos alocados e a data e hora.

```
{
  "id": 2,
  "courseId": 1,
  "classroomId": 1,
  "day": "Ter",
  "from": "10:00",
  "to": "12:00",
  "name": "PL1",
  "teacherId": 92001,
  "totalStudentsRegistered": 15
}
```

- **allocations**: representa a alocação de um estudante a um turno.

```
{
  "id": 1,
  "studentId": 104302,
  "shiftId": 30
}
```

### 3.4 Bibliotecas e Ferramentas Utilizadas

- **Vue 3 + Vite**: base da aplicação, permitindo desenvolvimento rápido com hot-reload, modularidade e performance.
- **Pinia**: biblioteca oficial de gestão de estado para Vue. Substitui o Vuex com uma abordagem mais simples e compatível com o Composition API.
- **Vue Router**: para navegação entre páginas, utilizando rotas dinâmicas com parâmetros (/messages/:userId).
- **TypeScript**: tipagem estática, garantindo maior robustez e previsibilidade no código.
- **json-server**: utilizado como mock backend para simular chamadas GET, POST, DELETE, etc., num ambiente RESTful local.
- **ESLint + Prettier**: ferramentas de linting e formatação automática de código, garantindo consistência no estilo de escrita.

### 3.5 Comunicação com o Backend

A comunicação com o backend é feita através de chamadas `fetch` diretas para endpoints REST simulados com `json-server`. As operações seguem os métodos tradicionais:

- GET para leitura de dados (/allocations, /messages, /shifts)
- POST para criação de novos registos
- DELETE para remoção de alocações antigas

O estado local é sincronizado com o servidor manualmente quando necessário, através de funções como `syncAllocations()` e `syncChanges()`.

## 4 Reflexão sobre a Aplicação Desenvolvida na Perspetiva do Utilizador

A aplicação desenvolvida tem como principal objetivo permitir a gestão de trocas de turnos entre estudantes de forma intuitiva, rápida e eficiente. Tendo em conta a perspetiva do utilizador, foram identificados diversos pontos fortes e algumas limitações, que se discutem de seguida.

### 4.1 Pontos Fortes

- **Interface clara e responsiva:** A utilização do Vue 3 em conjunto com boas práticas de organização de componentes permitiu desenvolver uma interface reativa e responsiva. As transições entre páginas e a atualização dos dados ocorrem de forma fluída, com feedback visual imediato ao utilizador.
- **Simplicidade nas ações:** O processo de troca de turnos foi desenhado para ser direto e intuitivo. Os utilizadores conseguem enviar pedidos de troca ou remoção de forma simples, sem passos intermediários confusos.
- **Feedback visual consistente:** Os botões e ícones de ação mantêm-se visíveis mesmo quando as ações não estão disponíveis, apresentando-se atenuados (*grayed out*). Esta abordagem melhora a previsibilidade e evita situações de frustração, onde o utilizador não entende por que razão um botão desapareceu.
- **Separação clara das mensagens:** A divisão entre mensagens recebidas, enviadas e lixo ajuda o utilizador a localizar rapidamente os pedidos relevantes, reduzindo a complexidade de navegação.
- **Estrutura modular e escalável:** A separação em *stores* e componentes reutilizáveis permite uma manutenção mais fácil da aplicação e futuras extensões com novas funcionalidades.

### 4.2 Pontos Fracos

Apesar dos pontos positivos, existem ainda áreas que podem ser melhoradas:

- **Contraste insuficiente de cores:** Um dos problemas mais relevantes identificado prende-se com a acessibilidade. Testes realizados com a ferramenta Pa11y revelaram que a aplicação falha em vários critérios de contraste de cor. Este problema tem origem no protótipo inicial, onde foram escolhidas combinações de cores com baixo contraste (ex: texto claro sobre fundos brancos ou azuis-claro). Este aspeto compromete a legibilidade da interface para utilizadores com dificuldades visuais e não respeita os níveis mínimos exigidos pelo WCAG 2.1.
- **Ausência de confirmações visuais em certas ações:** Embora a remoção de notificações de envio tenha simplificado a experiência, em alguns casos (como quando uma troca é aprovada) pode ser útil apresentar uma confirmação mínima para reforçar ao utilizador que a ação foi completada com sucesso.
- **Falta de mecanismos de filtro e ordenação:** Torna-se evidente a falta de alguns mecanismos de filtro ou ordenação, principalmente no âmbito de procura de estudantes. A aplicação não tem qualquer meio de procura de alunos por nome, semestre inscrito ou até número de aluno.
- **Dependência de sincronizações manuais:** Algumas ações exigem chamadas explícitas ao backend para sincronizar dados (*syncChanges*, *syncAllocations*). Embora estas operações funcionem corretamente, um sistema de sincronização automática ou em segundo plano traria maior transparência ao utilizador.

### **4.3 Testes de Acessibilidade com Pa11y**

Foi utilizado o analisador automático Pa11y para verificar o cumprimento de normas básicas de acessibilidade (WCAG 2.1). Os testes foram realizados sobre as principais páginas da aplicação, nomeadamente:

- Página de login
- Página de gestão administrativa (diretor do curso)
- Página de gestão de alocações
- Página do estudante
- Página de mensagens (inbox) e de criação de mensagens

Os resultados revelaram conformidade geral com os critérios técnicos, sem erros estruturais críticos. No entanto, destacou-se consistentemente o problema do baixo contraste entre cores, já referido anteriormente. Esta falha será documentada no repositório do projeto através dos relatórios exportados.

### **4.4 Considerações Finais**

A aplicação consegue cumprir o seu objetivo central de facilitar trocas de turno entre estudantes, com uma experiência de utilização simples, coesa e fluída. As principais oportunidades de melhoria prendem-se com a acessibilidade (contrastos) e algumas pequenas melhorias na experiência de interação. De futuro, seria também pertinente realizar testes de usabilidade com utilizadores reais para validar as decisões tomadas e recolher feedback qualitativo.

## 5 Manual de Utilização

De seguida, são apresentadas instruções sobre como utilizar a aplicação, incluindo uma explicação das ações possíveis, os seus efeitos, e uma descrição das páginas e popups disponíveis para os utilizadores.

### 5.1 Descrição das Páginas

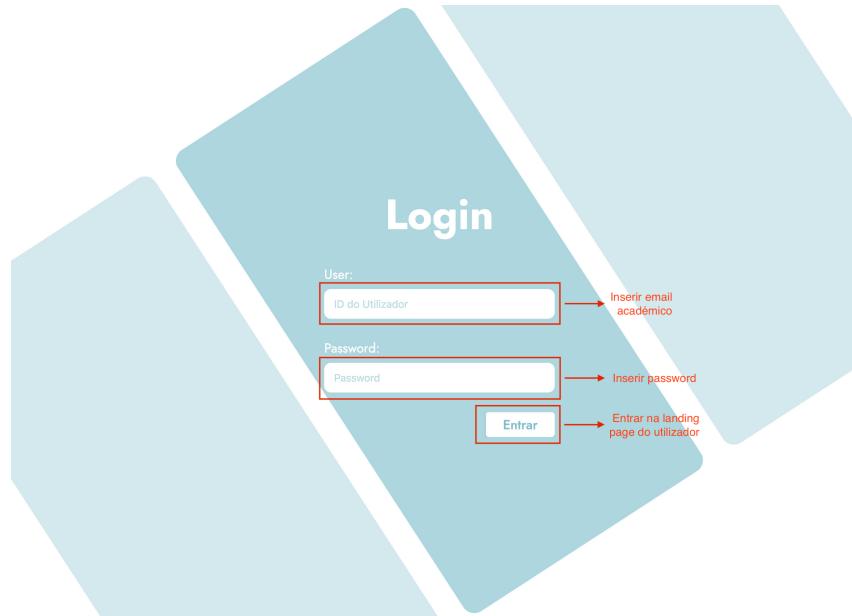


Figura 1: Página de Login

Ao aceder à aplicação, o utilizador é encaminhado para a página de login. Como ilustrado na figura, deve introduzir as suas credenciais nos campos apropriados. Caso estejam corretas, será redirecionado para a sua página principal.

Figura 2: Página do Diretor de Curso

Se o utilizador for um diretor de curso, poderá realizar várias ações conforme o seu objetivo. A figura apresenta os diferentes componentes com os quais pode interagir:

- Navegação** – O utilizador pode navegar para a página de mensagens clicando no ícone de envelope ou sair da aplicação através do botão de logout. O botão escurecido indica a rota atualmente ativa, estando desativado para prevenir a sua seleção.
- Filtros de horário** – É possível selecionar entre os 6 semestres disponíveis (de 1 a 6) e, posteriormente, escolher uma Unidade Curricular (UC) correspondente. O botão à direita permite limpar os filtros aplicados. O filtro de UC apenas fica disponível após a seleção de um semestre.
- Lista de Alunos** – O diretor pode clicar num aluno para abrir um popup com informações adicionais, descrito mais à frente neste relatório.
- Editar Alocações** – Clicando no ícone correspondente, o utilizador é redirecionado para a página de edição de alocações do aluno selecionado.
- Turnos no Horário** – Ao selecionar um turno no horário apresentado, são exibidos os detalhes desse turno específico.

Figura 3: Página de Edição de Alocações

Após escolher um aluno, o diretor de curso é redirecionado para a página de edição das suas alocações. Esta página mantém os mesmos mecanismos de navegação e filtragem, com as seguintes funcionalidades adicionais:

- Troca de Turnos** – O diretor pode clicar nos turnos atuais do aluno (o botão superior representa os turnos teóricos, o inferior os práticos), abrindo uma lista de turnos disponíveis para troca. A seleção de um turno reflete-se de forma reativa no estado do horário e dos turnos.
- Cancelar Operação** – Em qualquer momento, o utilizador pode cancelar a operação e regressar à página anterior clicando no botão "Voltar".
- Aplicar Alterações** – O botão "Aplicar" não confirma de imediato a alteração, mas abre um popup de confirmação para garantir a decisão do utilizador.

The screenshot shows the student dashboard with the following sections:

- Meu Horário (My Schedule):** A grid showing the student's schedule from 08:00 to 20:00 across five days (Seg, Ter, Qua, Qui, Sex). The grid includes course names like IO, MNOL, POO, RC, SO, and BD, along with their respective periods (T1, T2, PL1, PL2, PL3, PL4, PL5).
- Inscrições (Registrations):** A list of courses the student is registered in, with each course showing its status (N/A or Yes) and two circular icons.
- Top Bar:** Includes a search bar, a date selector, and three buttons labeled (1), (2), and (3).

Figura 4: Página do Aluno

Na página do aluno, muitas das funcionalidades são semelhantes às da página do diretor de curso, com algumas diferenças importantes. O aluno não necessita de aplicar filtros para visualizar o seu horário; este é carregado automaticamente com base nas UCs em que está inscrito.

- Navegação** – Funciona da mesma forma, com redirecionamentos ajustados ao perfil do aluno autenticado.
- Filtros** – Mantêm-se idênticos aos da página do diretor.
- Alternância de Horário** – O botão permite alternar entre "Horário Completo" e "Horário Atual". O primeiro mostra todos os turnos das UCs em que o aluno está inscrito; o segundo mostra apenas os turnos nos quais está efetivamente alocado.
- Pedido de Troca de Turno** – Ao clicar num turno no horário, é aberto um popup para o aluno requisitar a troca desse turno.

The screenshot shows the messaging interface with the following components:

- Mensagens (Messages):** The inbox view showing messages from Luis Gonçalves and Ana Gomes. One message from Luis Gonçalves is highlighted with a red border.
- Mensagens - Nova Mensagem (New Message):** The compose screen for sending a new message. It includes fields for 'Para:' (To), 'Assunto:' (Subject), a message body area, and an 'Enviar' (Send) button.
- Top Bar:** Includes a search bar, a date selector, and three buttons labeled (1), (2), and (3).

Figura 5: Páginas de Mensagens

Tanto o diretor de curso como os alunos têm acesso à página de mensagens. Esta inclui as seguintes funcionalidades:

1. **Navegação** – Mantém-se idêntica, com a secção de mensagens desativada (escurecida) por já estar ativa.
2. **Caixas de Mensagens** – O utilizador pode navegar entre a caixa de entrada, mensagens enviadas e mensagens apagadas.
3. **Nova Mensagem** – O botão ”Nova mensagem” redireciona para a página de composição de mensagens.
4. **Visualização Detalhada** – Ao clicar numa mensagem, esta é expandida para mostrar o conteúdo completo, bem como ações adicionais.
5. **Troca Rápida** – Caso a mensagem seja de ”troca rápida”, o diretor de curso pode aceitá-la ou rejeitá-la diretamente nesta página.
6. **Apagar a Mensagem** – O utilizador pode apagar a mensagem expandida, movendo-a para a caixa de mensagens apagadas.
7. **Responder a Mensagem** – O utilizador pode optar por responder a uma mensagem, sendo redirecionado para a página de composição, com os campos ”Para” e ”Assunto” já preenchidos.

## 5.2 Descrição dos Popups



Para os popups de **informações adicionais acerca do aluno** e **requisição de troca de turno**, o utilizador pode, respetivamente:

1. **Fehcar o Popup** - A qualquer momento, clicando no botão de fecho disponível, o utilizador pode fechar o popup. Esta funcionalidade está presente em todos os popups da aplicação, com efeito consistente.
2. **Contactar** - Redireciona o diretor de curso para a página de composição de mensagens, com o campo ”Para” já preenchido com o email do aluno em questão.

---

1. **Aderir** - Ação que envia automaticamente uma mensagem de ”troca rápida” para o diretor de curso, formalizando o pedido de troca de turno.



Para os popups de detalhes de turno, troca de sala e avisos, o utilizador pode, respetivamente:

1. **Editar Sala** - Ao clicar no ícone de edição, o estado do popup avança para o próximo estado (troca de sala).

---

1. **Recuar** - O utilizador pode, além de poder fechar o popup a qualquer momento, recuar do estado atual.
2. **Selecionar Sala** - seleciona a nova sala para o turno, o que abrirá um novo popup com um aviso de confirmação.

---

1. **Aceitar** - O utilizador aceita e reconhece o aviso.
2. **Recusar** - O utilizador recusa aviso e cancela a operação.

## 6 Reflexão sobre o Trabalho

Durante o desenvolvimento da aplicação, o grupo deparou-se com vários desafios técnicos e organizacionais, mas também com aprendizagens relevantes e uma percepção bastante positiva relativamente à utilização da *framework* Vue.

### Principais Dificuldades

A maior dificuldade enfrentada esteve relacionada com a implementação de um sistema de *stores* modular que conseguisse gerir corretamente o estado da aplicação e garantir sincronizações consistentes com a base de dados. Foi necessário pensar cuidadosamente na separação de responsabilidades entre os diferentes *stores* e na forma como estes interagem entre si. Adicionalmente, surgiram complexidades quando as ações de um *store* tinham implicações em outros — por exemplo, aprovar uma mensagem de troca de turno obriga à atualização do estado de várias entidades em simultâneo. Gerir esta coordenação e garantir que os dados se mantêm consistentes tanto no cliente como no servidor exigiu planeamento, testes frequentes e algumas refatorizações do código.

### Pontos Positivos e Opiniões do Vue

Apesar das dificuldades, a experiência de desenvolvimento com Vue.js foi altamente positiva. De forma geral, a biblioteca mostra-se bastante intuitiva e a curva de aprendizagem foi rápida com o apoio adicional à documentação oficial.

A estrutura reativa do Vue, aliada à simplicidade da sintaxe, permitiu desenvolver uma interface dinâmica com menos código do que seria necessário noutras soluções.

Além disso, a integração com ferramentas modernas como Vite, Pinia e Vue Router tornou o processo de desenvolvimento mais eficiente e modular. A funcionalidade de *hot reload*, por exemplo, reduziu consideravelmente o tempo de testes e ajustes durante a fase de implementação.