

**Aplicação Web para Calendário de Avaliações**

Licenciatura em Engenharia Informática

Alexandre Jorge Casaleiro dos Santos, 2181593

Miguel Calado Cerejo, 2192779

Leiria, julho de 2022



**Aplicação Web para Calendário de Avaliações**

Licenciatura em Engenharia Informática

Alexandre Jorge Casaleiro dos Santos, 2181593

Miguel Calado Cerejo, 2192779

Trabalho de Projeto da unidade curricular de Projeto Informático realizado sob a orientação do Professor Vítor Manuel de Oliveira Pegado de Noronha e Távora e do Professor Pedro Miguel Cardoso Gago.

Leiria, julho de 2022

# Dedicatória

Inserir aqui a dedicatória. Trata-se de um elemento **facultativo**.

Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória.

Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória.

# Agradecimentos

Inserir aqui os agradecimentos. Trata-se de um elemento **facultativo**.

Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos.

Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos.

# Resumo

Inserir aqui o resumo. Trata-se de um elemento **obrigatório**.

Deve começar sempre numa página ímpar. Se ocupar um número par de páginas (p. ex. 2), deve ajustar-se o texto para que a próxima secção (abstract) se inicie numa página ímpar. O resumo deve acabar com a lista de palavras-chave.

**No resumo deve dar-se nota das principais ideias do trabalho (objetivos e conclusões).**

Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo.

Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo.

**Palavras-chave:** máximo 6 palavras separadas por “,”

# Abstract

Please insert here the abstract in English. This is a **mandatory** element.

The abstract should always start in an odd page. If the length is a multiple of two, the text should be adjusted in order to the next section start also in an odd page. The abstract should end with a list of keywords.

Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English.

Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English.

**Keywords:** maximum of 6 words separated by “,”

Índice

Trata-se de um elemento **obrigatório**. Nota: **o índice nunca figura do índice.**

[Dedicatória ii](#_Toc107996566)

[Agradecimentos iii](#_Toc107996567)

[Resumo iv](#_Toc107996568)

[Abstract v](#_Toc107996569)

[Lista de Figuras x](#_Toc107996570)

[Lista de tabelas xi](#_Toc107996571)

[Lista de siglas e acrónimos xii](#_Toc107996572)

[1. Introdução 1](#_Toc107996573)

[2. Descrição do Problema 2](#_Toc107996574)

[3. Soluções para o Problema 4](#_Toc107996575)

[3.1. Tipos/Grupos de Utilizador 4](#_Toc107996576)

[4. Tecnologias Utilizadas 8](#_Toc107996577)

[4.1. Docker 8](#_Toc107996578)

[4.2. Laravel 9](#_Toc107996579)

[4.3. MySQL 10](#_Toc107996580)

[4.4. Redis 10](#_Toc107996581)

[4.5. React JS 10](#_Toc107996582)

[4.6. Diagrama de Arquitetura 11](#_Toc107996583)

[5. Reconhecimento do trabalho já existente 12](#_Toc107996584)

[5.1. Projeto do ano anterior 12](#_Toc107996585)

[5.2. Problemas encontrados 12](#_Toc107996586)

[5.3. Exemplos de problemas de usabilidade 13](#_Toc107996587)

[5.4. Novas funcionalidades 20](#_Toc107996588)

[6. Metodologia 23](#_Toc107996589)

[7. Trabalho Desenvolvido 25](#_Toc107996590)

[7.1. Base de Dados 25](#_Toc107996591)

[7.2. API 25](#_Toc107996592)

[7.3. Autenticação 26](#_Toc107996593)

[8. Trabalho a desenvolver no futuro 27](#_Toc107996594)

[8.1. Teste com vários utilizadores 27](#_Toc107996595)

[8.2. Agrupar avaliações 27](#_Toc107996596)

[8.3. Exportação de *Webservice* 27](#_Toc107996597)

[8.4. Traduções 27](#_Toc107996598)

[8.5. Filtros nas páginas 27](#_Toc107996599)

[8.6. Processo de Gestão dos Calendários 29](#_Toc107996600)

[8.6.1. Criação de Calendário 29](#_Toc107996601)

[8.6.2. Alteração de um Calendário Definitivo 31](#_Toc107996602)

[9. Validação do trabalho anteriormente realizado 33](#_Toc107996603)

[9.1. Título da secção 34](#_Toc107996604)

[10. Título do capítulo 35](#_Toc107996605)

[10.1. Título da secção 37](#_Toc107996606)

[10.2. Título da secção 37](#_Toc107996607)

[10.2.1. Título da subsecção 38](#_Toc107996608)

[10.2.2. Título da subsecção 38](#_Toc107996609)

[10.2.3. Título da subsecção 38](#_Toc107996610)

[10.3. Título da secção 38](#_Toc107996611)

[11. Título do capítulo 39](#_Toc107996612)

[12. Conclusões ou Conclusão 40](#_Toc107996613)

[Bibliografia ou Referências Bibliográficas 41](#_Toc107996614)

[Anexos 42](#_Toc107996615)

[Glossário 43](#_Toc107996616)

# Lista de Figuras

Elemento a figurar, **quando aplicável**.

[Figura 2.1 - Texto ilustrativo da figura 1. 3](file:///C:\ESTG\CCP_EI_2019_2020_2021\Projeto_Informatico\2020_2021\Docs_Moodle_Outros\Modelo_relatorio_projeto_ESTG.docx#_Toc92389045)

[Figura 2.2 - Texto ilustrativo da figura 2. 3](file:///C:\ESTG\CCP_EI_2019_2020_2021\Projeto_Informatico\2020_2021\Docs_Moodle_Outros\Modelo_relatorio_projeto_ESTG.docx#_Toc92389046)

# Lista de tabelas

Elemento a figurar, **quando aplicável**.

[Tabela 1.1 - Texto ilustrativo da tabela 1. 3](#_Toc92389035)

# Lista de siglas e acrónimos

Elemento a figurar, **quando aplicável**.

|  |  |
| --- | --- |
| API | Application Programming Interface |
| BD | Base de Dados |
| CCP | Comissão Científico-Pedagógica |
| CP | Conselho Pedagógico |
| ID | Identidade |
| ESTG | Escola Superior de Tecnologia e Gestão |
| GOP | Gabinete de Organização Pedagógica |
| IPLeiria | Instituto Politécnico de Leiria |
| LDAP | Lightweight Directory Access Protocol |
| REST | Representational State Transfer |
| UC | Unidade Curricular |
| UI | User Interface |
| VM | Virtual Machine (Máquina Virtual) |

# Introdução

Introdução deve conter resumidamente os seguintes elementos:

* O objeto do trabalho (o tema);
* A justificação ou a pertinência do tema;
* Os objetivos do trabalho (gerais e específicos, perguntas a responder ou hipóteses a testar);
* Os métodos e as técnicas utilizados;
* Estrutura do trabalho.

# Descrição do Problema

O calendário de avaliação de um curso é, segundo o regulamento da ESTG, o documento que estabelece os momentos de avaliação para aplicação dos métodos de avaliação, bem como as datas de divulgação de enunciados de trabalhos e de projetos. Neste sentido, é um documento de extrema importância tanto para professores como alunos, pois é através dele que podem ser consultadas as datas de realização dos vários momentos.

As diferentes Unidades de Ensino podem ter maneiras diferentes de criar os seus calendários, pelo que para este projeto foi usado o caso ESTG como exemplo.

Neste momento este é um processo moroso e muito sujeito ao erro humano, pelo que carece de várias fases de avaliação para garantir que no final o calendário é aprovado e disponibilizado à comunidade académica e não necessita de mais alterações. Para isso, o processo de criação de um calendário de avaliação atualmente é:

Inicialmente a direção da escola aprova o calendário escolar do ano letivo, definindo os períodos das várias épocas de avaliação, das atividades letivas e das interrupções. Com base nessa informação, o Gabinete de Orientação Pedagógica (GOP) efetua semestralmente a criação dos calendários de avaliação de cada curso, incluindo as datas de avaliações de algumas Unidades Curriculares (UCs) que sejam partilhadas por vários cursos (como o caso de Inglês e Matemáticas).

De seguida os calendários são remetidos para os Coordenadores de Curso, que passam a ser responsáveis por apresentar a proposta de calendário do seu curso. O Coordenador de Curso poderá solicitar aos responsáveis de cada UC o preenchimento das avaliações referentes à sua UC, ou optar por ser ele a marcar todas as datas de avaliação do seu curso. Caso um Coordenador necessite da ajuda dos responsáveis das UCs, deve no final voltar a rever o calendário completo, para garantir que não existem erros ou duplicação de informação.

A partir deste momento os calendários são normalmente divulgados junto dos alunos, de forma temporária, para que estes possam fazer algum comentário que julguem necessário em relação às datas das avaliações previamente marcadas. Depois de um período para a análise do calendário, os alunos e docentes pertencentes à Comissão Científico-Pedagógica (CCP) analisam também os comentários dos restantes alunos do curso, e podem então sugerir alterações ao calendário. Estas sugestões serão mais uma vez avaliadas pelo Coordenador de Curso, que fará então as alterações que achar necessárias para que o calendário possa então seguir para avaliação superior e posterior aprovação.

Neste processo, os calendários devem ser aprovados pelo GOP. Caso não seja aprovado, o Coordenador deve fazer as alterações necessárias para que o GOP aprove. Quando é aprovado, o calendário é então enviado para o Conselho Pedagógico (CP) da escola, que deve emitir um parecer em relação ao calendário apresentado.

Caso o CP encontre problemas no calendário, o Coordenador de Curso é chamado para rever novamente os comentários do CP, e submeter novamente o calendário para o GOP. Caso não existam problemas o calendário é logo enviado para o GOP, sendo que a partir deste momento o GOP tem total liberdade para editar diretamente o calendário caso assim seja necessário.

Finalmente, após este longo processo, o GOP submete o calendário à Direção da Escola, que deverá aprovar o calendário. Em caso de necessidade, o GOP fará os ajustes necessários para que o calendário seja aprovado pela Direção. Quando estiver aprovado, o GOP publica então os calendários na página da escola, sendo a partir deste momento considerados calendários definitivos, e estando públicos para quem os quiser consultar.

Neste momento, o processo de criação de calendários de avaliação está completo, no entanto existe ainda a possibilidade de um calendário necessitar de alterações até ao final do semestre. Para isso, e com os motivos devidamente justificados para tal, o Coordenador de Curso, em articulação com o responsável da UC, deve criar um calendário temporário, e submeter a proposta de alteração do calendário para o GOP, que deverá validar e submeter novamente à Direção da Escola, de forma a ser novamente aprovado, sendo novamente publicado pelo GOP de seguida.

No final, apresentamos em anexo, o fluxo em modelo BPMN (*Business Process Model and Notation)* que demonstra visualmente o processo descrito acima.

# Soluções para o Problema

Tendo em conta o processo descrito anteriormente, que como é possível observar, é um processo muito complexo e que necessita de muitas validações para garantir que não há sobreposições ou duplicações de avaliações, o objetivo deste projeto é criar uma ferramenta que permita tornar este processo mais simples e mais automático.

Desta forma, promovendo maior consistência e garantia de cumprimento das regras necessárias para a criação de um calendário, maior rapidez desde o processo de criação até à sua aprovação e publicação, e ao mesmo tempo garantir uma certa flexibilidade para todos os utilizadores envolvidos neste processo, e para as várias Unidades de Ensino do IPLeiria.

Esta ferramenta, foi escolhida para ser construída na forma de uma aplicação Web, que já tem vindo a ser construída nos últimos dois anos, por dois outros grupos de alunos, sendo que o primeiro grupo acabou por fazer mais um levantamento de requisitos e começou a estruturar a BD da aplicação.

O segundo grupo, com base no trabalho anterior, conseguiu já implementar a maior parte das funcionalidades base, no entanto faltavam muitas validações e mensagens de erro, pelo que a aplicação não era prática de usar, e os professores tinham alguma dificuldade em conseguir testar a aplicação.

O nosso objetivo, como terceiro grupo, é melhorar a usabilidade da aplicação, e implementar as funcionalidades que o grupo anterior não conseguir, de forma a conseguirmos disponibilizar a aplicação e que os professores e outros utilizadores possam começar a testar e usar a aplicação, e resolver efetivamente o problema existente atualmente.

## Tipos/Grupos de Utilizador

Com base no levantamento de requisitos dos grupos anteriores, validaram-se os tipos/grupos de utilizadores envolvidos no processo e definiram-se as permissões que cada um poderá realizar.

Ainda assim, aplicação está construída de forma que estas permissões possam ser facilmente alteradas no futuro.

**Administrador de Sistema:**

O Administrador de Sistema deve poder ter permissões para tudo, no entanto o seu principal papel será o de gerir a parte auxiliar do processo, ou seja, efetuar todas as configurações necessárias para o bom funcionamento da aplicação.

O Administrador de Sistema deve poder proceder à configuração de:

|  |  |
| --- | --- |
| **Gerais** | **Administração** |
| Calendário Escolar  Unidades Curriculares;  Agrupamentos;  Cursos; | Anos Letivos  Unidades de Ensino;  Fases de Calendário;  Tipos de Avaliação;  Tipos de Interrupções;  Grupos de Utilizadores;  Utilizadores; |

**Direção:**

A Direção de cada Escola / Unidade de Ensino deve poder proceder à consulta e aprovação definitiva de qualquer calendário de avaliação, da sua escola, independente das suas particularidades.

**Gabinete de Organização Pedagógica:**

O GOP tem a responsabilidade operacional de gerir os calendários de avaliação:

* Criar, atualizar e eliminar os Calendários de Avaliação;
* Agrupar UCs comuns a vários cursos;
* Marcar as avaliações das UCs comuns a vários cursos;
* Adicionar/editar/remover interrupções do calendário;
* Publicar calendários de avaliação definitivos;

**Coordenador de Curso:**

O Coordenador de Curso é responsável pela gestão e organização do calendário de avaliações do seu curso, em articulação (ou não) com os professores das UCs, e por levar à aprovação do GOP.

* Marcação e consultar as avaliações;
* Adicionar ou esconder comentários;
* Adicionar/editar/remover interrupções do calendário;
* Mudar a fase do calendário;
* Publicar o Calendário de Avaliação provisório do curso

**Responsável Unidade Curricular**

O responsável por cada UC deverá ter a possibilidade de:

* Adicionar e editar os Métodos de Avaliação da sua UC para cada época;
* Marcar avaliações nos calendários (caso o coordenador permita);
* Consultar os calendários de avaliação;

**Comissão Cientifica-Pedagógica:**

A CCP deverá ter a possibilidade de:

* Consultar os calendários de avaliação do seu curso;
* Adicionar ou esconder comentários;

**Conselho Pedagógico:**

O Conselho Pedagógico, após disponibilização dos Calendários provisórios pelo Coordenador de Curso, é lhe atribuída a responsabilidade de:

* Consultar os calendários de avaliação do seu curso;
* Adicionar ou esconder comentários;
* Registar o parecer do órgão sobre cada Calendário;

**Aluno:**

Numa fase final da elaboração dos Calendários de Avaliação, e antes destes serem definitivos, os alunos podem:

* Consultar os calendários de avaliação do seu curso;
* Adicionar comentários;

**Todos:**

Todos os utilizadores com autenticação efetuada podem:

* Consultar os calendários definitivos de qualquer curso, desde que esteja publicado como definitivo;

# Reconhecimento do trabalho já existente

A primeira fase do trabalho foi efetivamente fazer um reconhecimento de todo o trabalho já existente, e das tecnologias utilizadas (explicadas num capítulo mais à frente). Neste capítulo abordámos mais em detalhe alguns dos problemas que encontrámos.

## Projeto do ano anterior

Ao iniciar o projeto, e antes de começar a mexer no código, analisámos várias vezes o relatório e os vários diagramas/fluxos deixados pelo grupo anterior, e a testar a aplicação disponibilizada na VM da escola, de forma a entender os conceitos do negócio e podermos esclarecer algumas dúvidas com os professores.

Após essa primeira fase, começámos efetivamente a tentar trabalhar e perceber o código que tinha sido disponibilizado. Infelizmente esta fase não começou muito bem, pois não conseguíamos instalar o projeto fornecido na nossa máquina. Pelo que a primeira tarefa de todas foi a criação de um novo projeto, e aproveitámos para realizar a atualização para as novas versões das várias tecnologias utilizadas (*Laravel 9* e *React.js 17*).

Este processo demorou algum tempo até termos a certeza que todo o projeto estava a funcionar da mesma forma que na máquina virtual (uma vez que a atualização trouxe algumas *breaking changes* ao código inicial e para nós *React* era também uma novidade) e podermos então proceder a identificar os problemas e funcionalidades em falta do trabalho desenvolvido pelo grupo anterior.

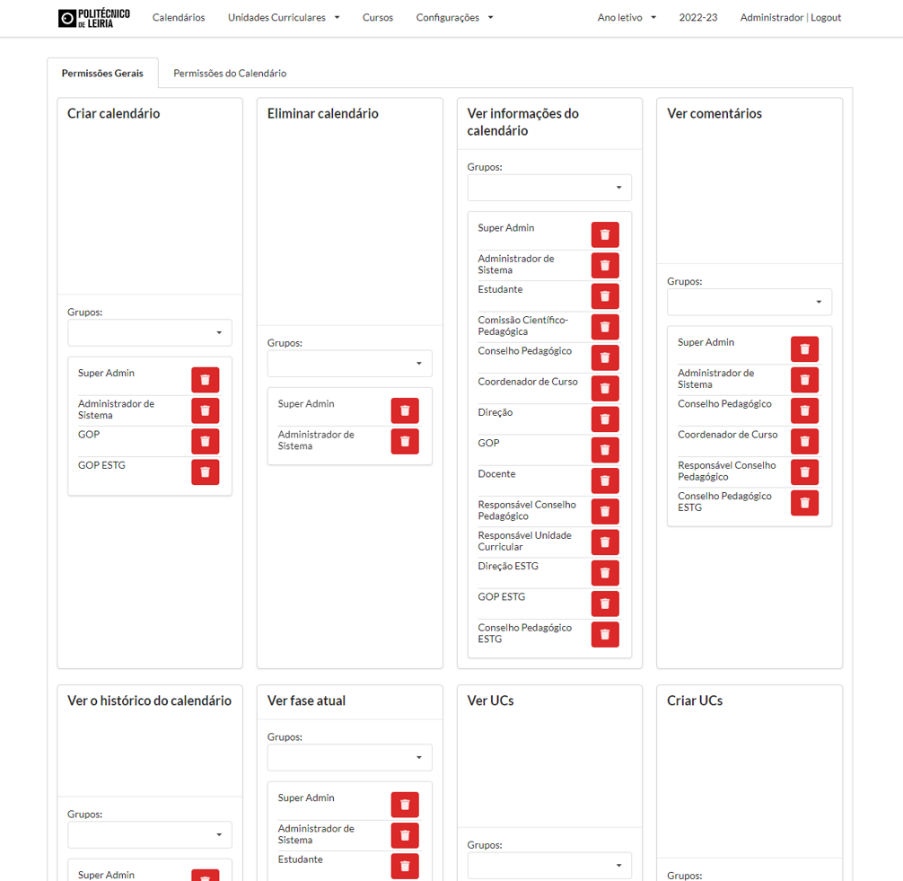
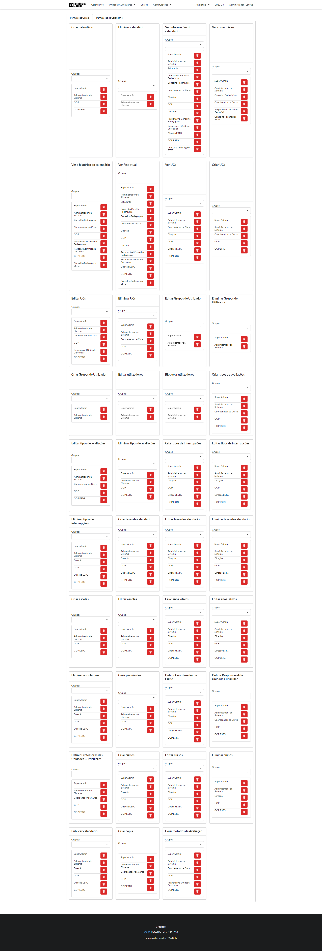
## Problemas encontrados

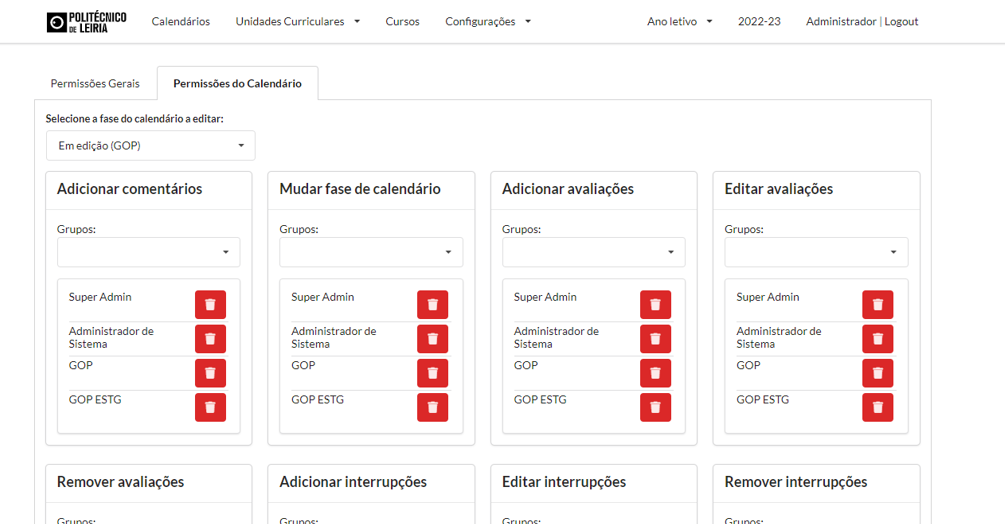
O projeto anterior estava a funcionar, e foi possível perceber que o grupo anterior até realizou um grande trabalho no desenvolvimento do projeto (uma vez que refizeram o código todo de raiz). No entanto, entre algumas funcionalidades e erros previamente identificados no relatório ou pelos professores, encontrámos os seguintes problemas na aplicação:

* Falta de validações e mensagens de erro quando alguma ação não funcionava;
* Código *hardcoded* tanto no Frontend como no Backend;
* Falta de filtros nas páginas de listagens;
* Código e BD não estruturados para possibilidade de a aplicação ter duplo idioma;
* BD com alguns problemas estruturais e dados incorretos ou em falta;
* *Migrations* e *seeders* em falta ou incompletos;
* Falta de documentação do projeto;
* Falta de testes às permissões, usando vários utilizadores;
* Falta de organização e gestão do código – código misturado em Português e Inglês (não resolvemos este problema, pois implicaria refazer o código todo existente);
* Usabilidade da aplicação – falta de praticidade na utilização de algumas ferramentas/páginas da aplicação (destacamos alguns destes erros de seguida);

Além disso, durante o decorrer do projeto, tivemos também alguns problemas com a conexão ao serviço do LDAP e utilização/sincronização do *WebService* dos cursos, na VM da escola.

## Exemplos de problemas de usabilidade

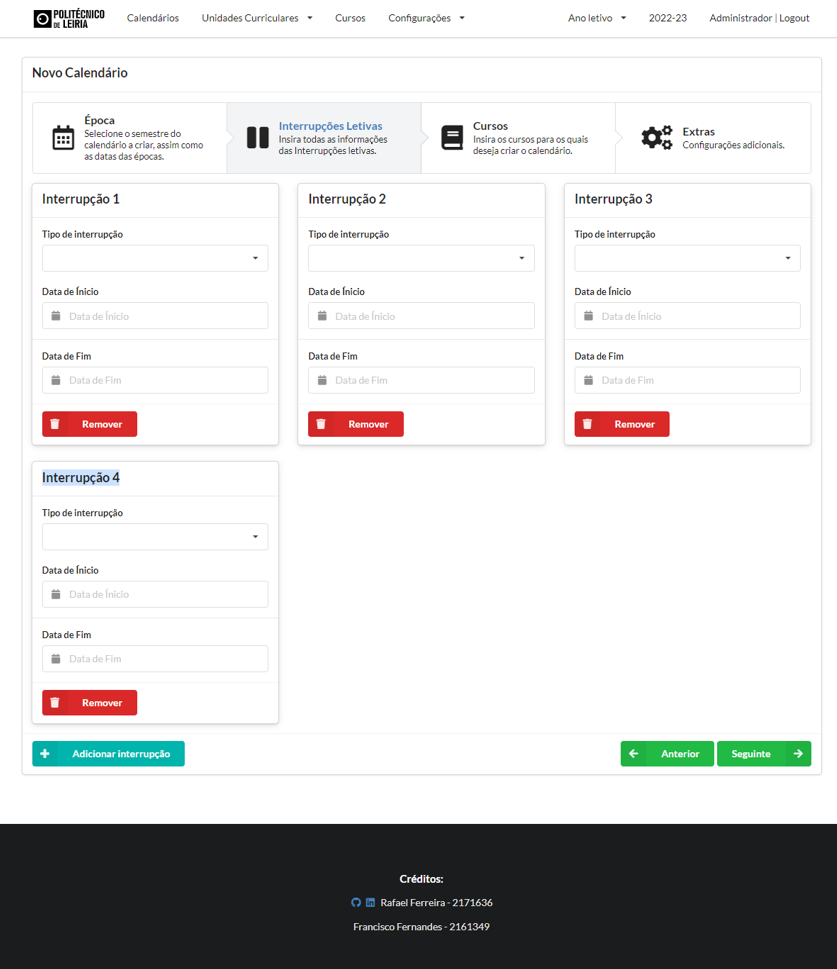
Gestão de permissões:

* Página muito comprida, muito difícil de ler e de gerir;
* Por cada interação era feito um pedido e era logo gravado na BD;
* Se uma das permissões tivesse muitos grupos associados e as permissões ao lado não, ficava uma página visualmente muito disforme;
* Quando apagávamos era necessário fazer *refresh* pois não removia da lista;
* Nas fases, não era possível ver todas as fases em simultâneo, pelo que ainda dificultava mais a gestão das permissões associadas a cada fase;

Uma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

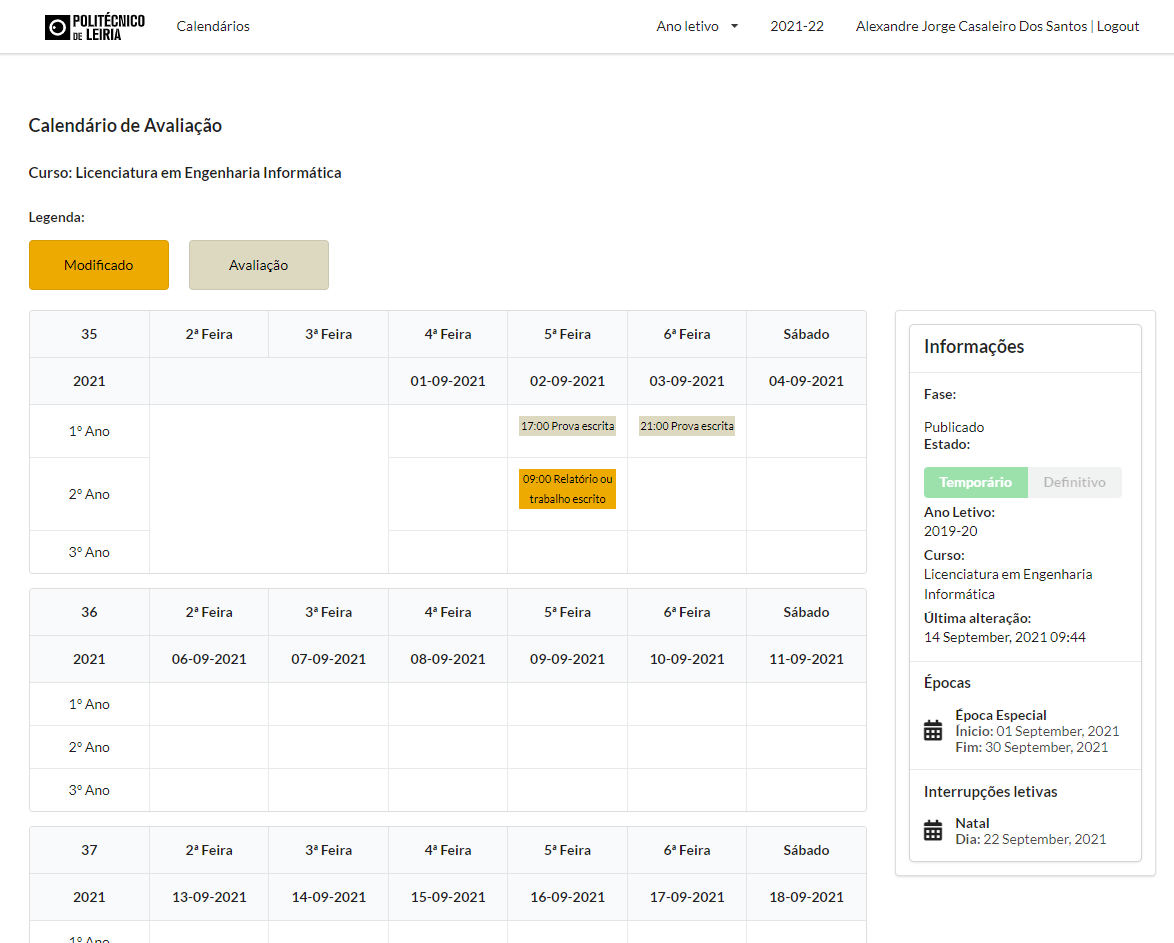
Descrição gerada automaticamenteCriação de calendário:



* Criação de interrupções pouco inteligente;
  + Existia um primeiro passo em que podia ser feita a introdução manual das interrupções, e havia um segundo passo em que podia ser feita uma importação automática das interrupções (feriados), no entanto era preciso saber que ela existia à partida, pois não havia indicações visuais disso;
  + Não era possível ter uma *preview* de quais as interrupções que a importação automática dos feriados iria criar. Apenas seria visível no detalhe do calendário depois de criado;
  + Se fosse necessário criar muitas interrupções, também facilmente iriam necessitar de muito espaço de ecrã e mais uma vez seria complicado de ler caso o ecrã fosse pequeno e fossem criadas muitas interrupções manualmente;
* Possibilidade de criar todos os cursos de uma vez. Ao selecionar todos os cursos não era possível entender que cursos estão a ser criados, no entanto não havia qualquer distinção entre cursos de licenciatura, mestrado ou TEsP, que normalmente têm datas diferentes (e por isso esta opção também não fazia sentido);
* Não havia qualquer tipo de validação visual nas datas marcadas, pelo que facilmente marcávamos a época de recurso em cima de outras épocas, que pelo regulamento não é possível;

Uma imagem com mesa

Descrição gerada automaticamenteDetalhe do calendário:

Uma imagem com texto

Descrição gerada automaticamenteUma imagem com mesa

Descrição gerada automaticamente

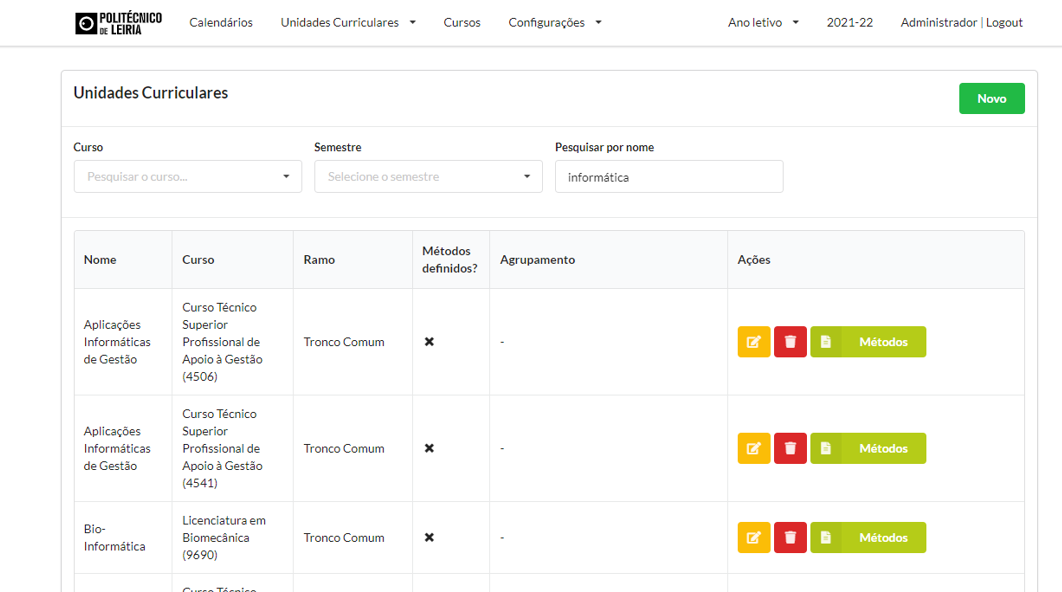
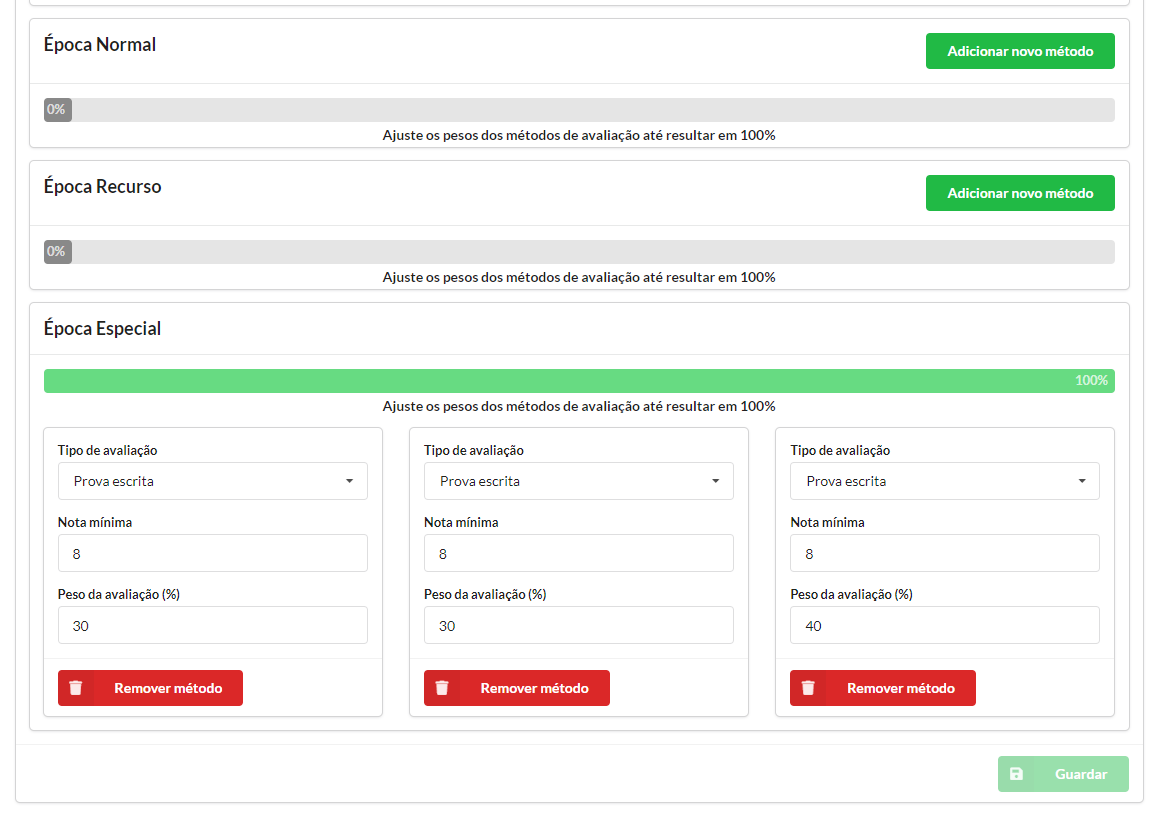
* Informações duplicadas ou desnecessárias a ocupar espaço, e informações importantes em falta;
* Espaço visualmente mal distribuído;
* Informações de ajuda não eram completamente relevantes para ajudar;
* Distinção entre utilizador normal (estudante) e professores/administradores quase inexistente, e era fácil desbloquear os inputs de forma estragar os calendários. Representava uma falta de segurança do lado do cliente;
* Sem forma de adicionar ou editar as interrupções;

Detalhe de UC e gestão de métodos de avaliação:

* Uma imagem com texto

  Descrição gerada automaticamentePossibilidade de editar os dados de uma UC. Estes dados devem vir sempre do WebService;
* Uma imagem com texto

  Descrição gerada automaticamenteNão possibilidade de gestão dos métodos. Listagem apenas para visualização;
* Não existe indicação de como proceder à gestão de métodos na página;



* Obrigatoriedade de preenchimento de todos os métodos;
* Utilização do espaço muito grande e torna-se confusa;
* Interação no preenchimento dos métodos confusa. Era necessário editar o peso de cada método para poder adicionar mais, uma vez que sempre que se adicionava um método, este ocupava toda a percentagem disponível. No entanto não existia indicação visual que era necessário editar o peso dos métodos para ser possível adicionar outro;

Uma imagem com texto

Descrição gerada automaticamente

## Novas funcionalidades

Durante os desenvolvimentos do projeto, foram identificadas e criadas algumas novas funcionalidades:

Gerais:

* Adicionados mais filtros nas várias páginas de listagens;
* Adicionadas traduções nas várias páginas, tanto a nível de Frontend como adicionados novos campos na BD de forma a permitir gestão a gestão dos dois idiomas na plataforma;
* Adicionado um plugin (Server Logs Viewer) para *logs* do servidor, para deteção e resolução de problemas na sincronização dos cursos/UCs na máquina virtual;
* Criada nova API para expor os dados das avaliações marcadas na plataforma para outras aplicações consumirem (apenas para calendários publicados);

Unidades Curriculares:

* Duplicação de métodos de avaliação entre épocas e automatização do processo de descrição dos métodos;
* Criação de um sistema de *logs* das alterações nas UCs (edições nos ramos, professores ou métodos);
* Ao definir uma UC com método de avaliação de projeto, adicionar automaticamente outros 2 métodos (lançamento de enunciado e apresentação oral) para marcação das datas destes momentos também no calendário;

Configurações de Administração:

* Gestão de anos letivos: selecionar quais os visíveis e qual o calendário por defeito, e forma de adicionar/editar/remover anos letivos melhorada, uma vez que nem sempre era possível criar ou editar;
* Adicionar forma de ativar a sincronização das UCs e Cursos por semestres de forma manual;
* Adicionada uma *flag* na BD para na gestão das fases de calendário, podermos definir quais as fases em que é necessário que todos os métodos de avaliação estejam preenchidos para que possam ser selecionadas.

Calendário:

* Adicionada possibilidade de adicionar, editar e remover interrupções no detalhe do calendário (apenas era possível na criação);
* Remoção de bloqueadores que restrinjam a plataforma na marcação das avaliações, criando apenas avisos quando alguma ação que está a ser executada poderá ser errada (exemplo: adicionar duas vezes a mesma avaliação);
* Ferramenta de revisão do calendário, de forma a validar se todos os métodos/elementos de avaliação foram ou não colocados no calendário;
* Adicionada possibilidade de marcar a 10ª semana de aulas, sendo pré-calculada automaticamente;
* Adicionada possibilidade de uma avaliação ser realizada em aula, sendo que não necessita de indicação da hora ou sala, e que normalmente é realizada ao longo de 1 semana e não apenas num dia específico, pelo que foi também colocada a possibilidade de uma avaliação começar e terminar em dias diferentes;
* Criada distinção visual das épocas, quando existe possibilidade de haver mais do que uma época em simultâneo. Com isto atualização para automatizar a seleção da época no processo de marcação de avaliações;

Além destas, foram sendo identificadas mais algumas pequenas melhorias em funcionalidades já existentes que foram sendo realizadas, mas não tomámos nota, e outras funcionalidades, mas que por uma questão de gestão de tempo apontámos e serão mencionadas mais à frente numa secção para trabalho que poderá ser realizado no futuro para melhorar o projeto.

# Metodologia

O projeto foi desenvolvido com base numa metodologia *Agile* baseada nos princípios dos *Scrum*. Ou seja, não seguimos todas as “regras” do Scrum, mas marcámos reuniões de projeto semanais para em conjunto com os professores orientadores mostrarmos o trabalho desenvolvido durante essa semana e recebermos feedback de melhorias ou novas funcionalidades a implementar.

Durante cerca do primeiro mês e meio de trabalho fizemos um estudo mais intensivo do relatório e do projeto do ano anterior, analisando o código e a plataforma que tinha sido deixada numa máquina virtual (VM) da escola, tentando perceber bem os requisitos e conceitos de negócio do projeto e perceber quais as falhas existentes no projeto anterior, para fazer um ponto de situação. Neste período foi também onde tivemos de atualizar as tecnologias utilizadas (conforme explicamos mais à frente) e fazer alguns melhoramentos base de organização do código, na UI da aplicação e na performance das páginas.

No final deste período foi quando atualizámos a VM da escola pela primeira vez para que pudessem ser realizados os primeiros testes à aplicação. A partir daqui tentámos fazer lançamentos semanais da aplicação para a VM da escola, de forma que fossemos introduzindo as novas funcionalidades ou correções e pudéssemos receber feedback o mais rapidamente possível.

Ainda ao início do projeto, decidimos criar um quadro na plataforma *Trello*, que nos pudesse ajudar a gerir o projeto e as tarefas pendentes. No entanto com o decorrer do projeto, e há medida que íamos recebendo feedback, acabámos por passar a fazer essa gestão diretamente no projeto através de alguns ficheiros *markdown* dedicados exclusivamente a guardar o feedback das reuniões para garantir que não deixávamos nenhum ponto ficar esquecido.

De forma a garantir que todo o código se encontrava guardado e em segurança, tanto versões anteriores como atuais, foi usado o *GitHub*, sendo que sempre que disponibilizávamos uma versão nova na VM fazíamos também uma nova *release* do projeto no *GitHub*.

# Tecnologias Utilizadas

As tecnologias utilizadas foram escolhidas pelo grupo anterior. De seguida explicamos mais em detalhe cada uma delas e o que fizemos em relação ao código fonte inicial.

## Docker

O Docker é uma ferramenta muito utilizada hoje em dia, pois permite a criação dos chamados “*containers*” (contentores), que representam imagens virtuais de software estando isoladas e prontas a usar, tornando o processo de distribuição/partilha do código fonte de uma forma muito mais rápida entre máquinas/ambientes (desenvolvimento, testes ou produção).

Cada contentor, traz todo o software, bibliotecas e ficheiros de configuração necessários para funcionar tal e qual como estava no momento e na máquina em que foi criado. Como todos os contentores partilham os serviços de um único *kernel* do sistema operativo, são usados poucos recursos da máquina virtual.

Graças a isto, o Docker vem trazer uma grande rapidez na instalação e configuração de projetos em cada máquina. Basta ter o projeto no computador e correr com o Docker o projeto que está a funcionar como noutro computador qualquer.

Infelizmente, quando iniciamos o projeto, houve alguns problemas com a imagem do Docker que queríamos iniciar, e não conseguíamos colocar o projeto a funcionar corretamente nas nossas máquinas. Sendo assim, acabámos por recomeçar a utilização do Docker através do Laravel Sail, que já trás tudo configurado para trabalhar com o Docker.

A partir do momento que fizemos isto, correu tudo como esperado ao utilizar o Docker. Como trabalhamos em ambiente Windows, é necessário fazer as configurações do Ubuntu e WSL/WSL2 na nossa máquina inicialmente, e após essa configuração inicial, apenas necessitamos de correr dois comandos para iniciar a máquina.

Na tabela seguinte mostramos os comandos necessários para iniciar o container com toda a parte do servidor:

|  |
| --- |
| Através da consola do Ubuntu, e estando na pasta do projeto:  **…/project-root$** alias sail='[ -f sail ] && bash sail || bash vendor/bin/sail'  **…/project-root$** sail up -d |

## Laravel

O *Laravel* é uma framework PHP para o desenvolvimento de aplicações Web com o modelo MVC (*Model-View-Controller*). Um dos pontos fortes do *Laravel* é a sua gestão de pacotes, que é modelar com um sistema de dependências dedicado, as suas várias maneiras de acesso a BD’s relacionais e serviços que ajudam no desenvolvimento e manutenção da aplicação.

Na aplicação, o *Laravel* é usado para toda a lógica da parte do servidor, gerindo as rotas, ligações a APIs externas, sincronização dos cursos (e UCs e professores das UCs), e na implementação de uma API para consumo da aplicação *Front-end* feita com base em *React*. Houve ainda alguns pedidos para criação de uma API para comunicação com outras aplicações que possam vir a ser desenvolvidas, neste caso, que disponibilize as avaliações calendarizadas no nosso sistema para fora.

O *Laravel* controla ainda a autenticação na plataforma, tendo alguns utilizadores por defeito que funcionam com base no *Laravel Passport*, e fazendo sincronização com o serviço LDAP da escola.

Quando iniciámos o projeto, esta parte já estava bastante avançada, no entanto (como explicámos anteriormente), houve alguns problemas com o código inicial que não conseguíamos colocar a funcionar, pelo que optámos por atualizar para o *Laravel* 9 de forma a resolver esses problemas.

Além disso, havia algumas falhas que fomos corrigindo à medida que íamos entregando novas funcionalidades ou correções aos professores, incluindo correções de validações, melhorias de performance (especialmente na sincronização de UCs e Cursos), reorganização e melhoria das *migrations* e *seeders* da BD, e divisão e organização do código em módulos específicos, tendo em conta as boas práticas do *Laravel*.

## MySQL

O MySQL é um sistema de gestão de base de dados. Utiliza como base a linguagem SQL, que permite criar, modificar e extrair data de uma BD relacional, bem como controlar o acesso por utilizador à BD. As razões para usar o MySQL foi por ser rápido, fácil de usar, portabilidade, segurança, ocupar pouco espaço e recursos, e por ser grátis.

Neste caso não fizemos grandes mudanças à estrutura inicial, sendo que o nosso trabalho no MySQL foi principalmente algumas alterações à estrutura inicial do grupo anterior, para novas funcionalidades que foram sendo pedidas, e especialmente para as traduções. De qualquer forma, este trabalho foi feito através das *Migrations* do *Laravel*, de forma a ser fácil a recriação da BD em qualquer máquina de desenvolvimento ou na VM da escola.

## Redis

O *Redis* é uma estrutura de armazenamento rápido, que vem incluída por defeito no *Laravel Sail,* e que funciona como memória cache de forma a aumentar o desempenho de *queries* ou alguns pedidos.

Sendo também *open-source*, e vindo incluído por defeito com o *Laravel*, é bastante fácil de usar, incluindo tarefas agendadas (*cron jobs*) e filas (*queues*). No nosso caso, usamos para melhorar a performance da sincronização dos cursos e UCs.

## React JS

O React é uma biblioteca JavaScript de *open-source,* que tem o objetivo de criar interfaces de páginas web, num modelo Single Page Application (SPA), criando aplicações com uma grande performance. Foi lançado em 2013, e é mantido pelo Facebook, Instagram, outras empresas.

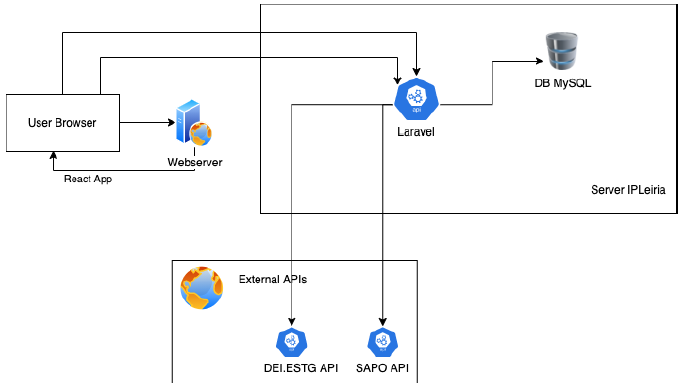
Como nas outras tecnologias, esta foi também outra herança da escolha do grupo anterior. Segundo eles: *“foi escolhido o React por esta ser uma biblioteca extremamente estável, assim como uma grande comunidade de suporte. Além disso, o React, comparando com o Vue.js, tem mais bibliotecas e ferramentas, e é mais simples e rápido na construção de uma aplicação complexa”*.

Para nós, visto esta ser uma tecnologia bastante conhecida e usada, e que não é abordada no curso de Engenharia Informática da ESTG do IPLeiria, esta era também uma novidade para nós. O nosso objetivo era efetivamente aproveitar este projeto para aprender a trabalhar com esta nova tecnologia. Este objetivo foi cumprido, no entanto não foi fácil quebrar a barreira inicial de conhecer a sintaxe e as diferenças que existem entre o React e o Vue, e atrasou um pouco o nosso planeamento inicial, pois existem muito mais diferenças entre as duas *frameworks* do que imaginávamos.

A maior parte do nosso trabalho foi executada nesta tecnologia, uma vez que é aqui que efetivamente está o *core* da aplicação que é usada pelos utilizadores, e além de rever e validar todo o trabalho que estava para trás, foi necessário desenvolver novas funcionalidades, e corrigir funcionalidades que não tinham ficado completas ou que não estavam realmente bem. Este foi um processo longo, ao qual foi necessário validar várias vezes com os professores se o que estava desenvolvido estava bem ou se eram necessárias alterações.

## Diagrama de Arquitetura

Na figura seguinte é possível verificar como é que cada tecnologia se interliga na aplicação.

A rever:

# Trabalho Desenvolvido

Como já referido anteriormente, este projeto acaba por ser a continuação de um trabalho realizado no ano anterior. No entanto, por causa de todos os problemas referidos, foi decidido fazer uma nova implementação, tanto a parte de Backend como de Frontend.

## Base de Dados

Durante todo o desenvolvimento a base de dados foi sendo alterada para acomodar os novos requisitos e ideias que foram aparecendo ao longo do projeto. Isto foi possível porque a BD foi toda construída seguindo as regras para fazer um modelo relacional.

A BD tem como principais tabelas os Utilizadores, as Unidades Curriculares e os Calendários de Avaliação.

Para dar suporte a estas tabelas, as Unidades Curriculares têm um Curso, que por sua vez tem uma Unidade de Ensino. Os Utilizadores têm um Grupo associado, e esse Grupo tem Permissões associadas, seja para Fases de Calendário como para acessos na aplicação. O Calendário de Avaliação tem por base as Avaliações, que tem uma Unidade Curricular associada.

Um diagrama completo da BD encontra-se no Anexo III – Base de Dados.

## API

Foi necessário desenvolver uma API totalmente nova. Esta nova API foi construída como sendo uma REST API.

Desta forma, foi-nos permitido controlar todas as rotas, mensagens de erro, e ter um maior controlo de acesso às chamadas da API.

Adaptando tudo o que estava feito anteriormente, foram utilizados os métodos corretos de acesso aos endpoints (POST; PATCH, DELETE; GET).

## Autenticação

A Autenticação é feita pelo serviço de LDAP da escola. Por isso, para se conseguir fazer login e aceder a qualquer funcionalidade da aplicação é necessário ter uma conta do Instituto Politécnico de Leiria (IPLeiria) válida.

Um Utilizador …

# Trabalho a desenvolver no futuro

Perante a impossibilidade de completar todo o trabalho do projeto, no presente capítulo, são descritas algumas sugestões de melhorias e de implementações.

## Teste com vários utilizadores

É necessário fazer um teste da aplicação, com 1 ou 2 cursos, para se perceber se existem mais necessidades e eventuais erros que possam existir.

## Agrupar avaliações

Existem épocas em que as avaliações de diferentes cursos são realizadas ao mesmo tempo, como acontece muitas vezes por exemplo em diurno e pós-laboral em Engenharia Informática.

Para isso seria necessário criar um agrupamento de avaliações, onde se referia que certa avaliação seria feita em conjunto com a mesma avaliação de outro curso.

## Exportação de *Webservice*

Criar uma exportação de *Webservice* os calendários do curso.

## Traduções

Por restrições temporais, não foi possível implementar as traduções da aplicação. A ideia desta funcionalidade é poder traduzir toda a aplicação para uma linguagem à escolha, e não apenas para certas palavras.

## Filtros nas páginas

Apesar das páginas já contarem com filtros, entendemos que poderiam ser adicionados mais filtros nas páginas.

**7.6. Alertas**

Adicionar mais detalhes nos alertas enviados pela API para que o utilizador comum entenda se aconteceu algum problema com a aplicação.

Ao longo da execução deste projeto foram também encontradas novas funcionalidades e requisitos.

* Autenticação com o serviço LDAP da escola
* Importação de dados através de Webservices já existentes.
  + Importação de cursos para o ano letivo corrente
  + Importação das UC’s dos cursos
  + Importação das inscrições às UC’s dos alunos
* Existirem 2 calendários em simultâneo
  + Face à constante alteração dos calendários, é necessário ter sempre um calendário que seja possível de alterar, mas que, só esteja visível para os intervenientes autorizados. Com esta funcionalidade, é possível ter 2 calendários em simultâneo do mesmo curso, mas um está em edição e o outro já foi disponibilizado a toda a comunidade académica, após passar por todo o processo de construção do calendário.
* Alertas aos alunos de alterações no calendário do seu curso.
  + Após a publicação de um calendário mais atualizado, os alunos poderão ver quais foram as modificações que ocorreram em relação à versão anterior do calendário, estando as diferenças claramente identificadas.

A rever:

## Processo de Gestão dos Calendários

O seguinte processo é o caso atual na realização dos calendários. Este é o workflow idealizado, no entanto este processo pode ser facilmente alterado com a alteração das permissões para os grupos pretendidos.

Para o todo o próximo processo funcionar fluidamente é necessário já ter um ano letivo criado, com os cursos e unidades curriculares todas corretas, e com o agrupamento das UC’s efetuados, bem como os métodos de avaliações dos agrupamentos feitos. Os métodos de avaliação das restantes UC’s apenas podem ser preenchidos quando existir um calendário para a sua época, portanto, apenas após a criação do calendário, é necessário adicionar os métodos de avaliação dessas UC’s.

### Criação de Calendário

Verificar o Anexo I – Processo de Criação de um Calendário de Avaliação para ajudar a descrever o seguinte processo.

Após o ano letivo estar criado, o GOP terá de criar um novo calendário. Aí tem de definir as datas de início e fim de cada época, as datas das interrupções e quais os cursos para criar o calendário. Cria o calendário e ele encontra-se na fase de “Criado”. Após isso o GOP altera a fase para “Em Edição (GOP)”.

Na fase “Em Edição (GOP)”, o GOP insere as avaliações das UC’s partilhadas. De seguida altera a fase para “Em Edição (Coordenador de Curso)”. No entanto, e apesar de ter passado à fase seguinte, o GOP pode a qualquer altura fazer uma alteração.

Na fase “Em Edição (Coordenador de Curso)”, o Coordenador de Curso tem a possibilidade de optar por uma das seguintes formas para a marcação das avaliações:

* Pode ser ele a agendar todos os momentos de avaliação do calendário e de seguida coloca o calendário na fase “Em Avaliação (Alunos)”.
* Pode realizar o calendário com a ajuda dos Responsáveis pelas UC’s e ser ele a preencher certas avaliações, depois passar a fase para “Em Edição (Responsável UC)”. Nesta fase tanto o Coordenador de Curso como os Responsáveis das UC’s podem agendar avaliações. Os Responsáveis das UC’s apenas podem marcar avaliações das UC’s pelas quais são responsáveis. Após as avaliações estarem marcadas e validadas pelo Coordenador de Curso, este passa a fase para “Em Avaliação (Alunos)”.

Na fase “Em Avaliação (Alunos)”, os alunos irão fazer os seus devidos comentários ao calendário. O calendário irá estar nesta fase durante um período de tempo a definir pelo Coordenador de Curso. Após esse tempo o Coordenador de Curso passa o calendário para a fase “Em Avaliação (CCP)”.

Na fase “Em Avaliação (CCP)”, a CCP filtra os comentários feitos pelos alunos, e faz os seus comentários com base nos comentários dos alunos. Após um tempo definido pelo Coordenador de Curso, este passa o calendário para “Em Edição (Coordenador de Curso)”.

Desta vez, na fase “Em Edição (Coordenador de Curso)”, o Coordenador de Curso faz ou não as alterações propostas pela CCP. Após fazer as alterações que achar necessárias, coloca o calendário na fase “Em Avaliação (GOP)”.

Na fase “Em Avaliação (GOP)”, o GOP valida as alterações realizadas. Se achar que são precisas alterações, pode colocar o calendário na fase “Em Edição (Coordenador de Curso)”, depois o Coordenador de Curso altera de acordo com essas alterações, e volta a colocar na fase “Em Avaliação (GOP)”. Este processo pode acontecer até o GOP considerar as alterações no calendário válidas. Quando estiver tudo correto para o GOP, este coloca o calendário na fase “Em Avaliação (Conselho Pedagógico)”.

Na fase “Em Avaliação (Conselho Pedagógico)”, o Conselho Pedagógico faz comentários sobre o calendário e emite um parecer sobre o mesmo. Quando acabar, tem 2 opções:

* Se existirem comentários, altera a fase do calendário para “Em Edição (Coordenador de Curso)”. Nessa fase, o Coordenador de Curso faz as alterações possíveis tendo em conta os comentários do Conselho Pedagógico e coloca o calendário na fase “Em Edição (GOP)”.
* Se não existirem comentários, altera a fase do calendário para “Em Edição (GOP)”.

Desta vez, na fase “Em Edição (GOP)”, o GOP pode fazer as alterações que achar pertinentes e coloca a fase do calendário como “Em Avaliação (Direção)”.

Na fase “Em Avaliação (Direção)”, a Direção valida o calendário. Após validar o calendário e aprová-lo, coloca o mesmo na fase de “Aprovado”. Não aprovando, volta a colocar o calendário na fase “Em Edição (GOP)”, e o processo repete-se até a Direção aprovar o Calendário.

Na fase “Aprovado”, o GOP apenas tem de publicar o calendário definitivo.

### Alteração de um Calendário Definitivo

Um Calendário de Avaliação corre sempre o risco de precisar de alterações. Para isso e após este já ser definitivo e ter sido publicado, existe uma forma de o alterar.

Verificar o Anexo II – Processo de Alteração de um Calendário de Avaliação para ajudar a descrever o seguinte processo.

Após ter sido publicado, o GOP ou o Coordenador de Curso podem criar uma cópia do calendário já publicado.

Após a criação, a fase do calendário irá para “Em Edição (Coordenador de Curso)”. O Coordenador de Curso realiza as alterações necessárias e coloca o calendário na fase “Em Avaliação (GOP)”.

Nesta fase de “Em Avaliação (GOP)”, o GOP verifica se está tudo correto.

* Não estando tudo correto, volta a colocar o calendário na fase “Em Edição (Coordenador de Curso)”. E repetem estes passos até estar correto. Após estar correto o GOP coloca o calendário na fase “Em Avaliação (Direção)”.
* Estando tudo correto, coloca o calendário na fase “Em Avaliação (Direção)”.

Tal como na criação, na fase “Em Avaliação (Direção)”, a Direção valida o calendário. Após validar o calendário e aprová-lo, coloca o mesmo na fase de “Aprovado”.

Na fase “Aprovado”, o GOP apenas tem de publicar a nova versão do calendário definitivo.

# Título do capítulo

Aqui, inicia-se o desenvolvimento dos trabalhos e deve indicar-se o título do capítulo primeiro.

O desenvolvimento do trabalho deve ser adequado à natureza da unidade curricular (dissertação/trabalho de projeto/relatório de estágio) e deve seguir as práticas mais disseminadas na área em causa.

Estrutura: pode ter, por exemplo, capítulos, secções e subsecções.

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

De seguida, apresenta-se um exemplo de como as imagens devem ser colocadas no texto:

* Centradas;
* Numeradas;
* Referenciadas no texto;
* Sem contornos;
* Eventualmente, colocar a(s) figura(s) numa tabela para melhorar a formatação;
* Usar sempre o estilo “caption” para o texto das legendas;
* Identificação da figura em “negrito” (o estilo trata desta parte). Texto da legenda não deve estar em negrito.

|  |
| --- |
| Figura 2.1 - Texto ilustrativo da figura 1. |

De seguida apresenta-se a figura 2.

|  |
| --- |
| Figura 2.2 - Texto ilustrativo da figura 2. |

Cuidados com a inserção de tabelas:

* Centradas;
* Numeradas;
* Referenciadas no texto;
* Usar sempre o estilo “caption” para o texto das legendas;
* Identificação da tabela em “negrito” (o estilo trata desta parte). Texto da legenda não deve estar em negrito.

Tabela 1.1 - Texto ilustrativo da tabela 1.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Continuação do texto do desenvolvimento.

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

## Título da secção

A figurar no caso de o capítulo ter várias secções. Nota: apenas devemos criar secções e subsecções quando existem mais do que uma.

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

## Título da secção

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

### Título da subsecção

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

### Título da subsecção

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

### Título da subsecção

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

## Título da secção

# Conclusões ou Conclusão

Inserir aqui as conclusões ou conclusão. Trata-se de um elemento **obrigatório**.

A conclusão:

* Deve ser sucinta;
* Não deve conter informações ou ideias novas;
* Deve permitir concluir se se atingiram os objetivos enunciados na introdução.

Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão.

Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão.

# Bibliografia ou Referências Bibliográficas

Inserir aqui a bibliografia ou referências bibliográficas. Trata-se de um elemento **obrigatório**.

Notas: o sistema a adotar para a apresentação das referências bibliográficas e as suas citações deve:

* Respeitar uma norma estabelecida;
* Seguir as práticas mais disseminadas na área em causa;
* Ser empregue de modo uniforme em todo o documento.

Bibliografia – quando se coloca toda a bibliografia consultada;

Referências bibliográficas – quando se faz referência apenas à bibliografia citada.

# Anexos

Elemento a figurar, **quando aplicável**.

# Glossário

Elemento a figurar, **quando aplicável**.