

Definição de estilo: Índice 1: Espaçamento entre linhas: 1,5 linhas

Afonso Boto 89285, Tomás Candeias 89123, Jorge Oliveira 84983,
Afonso Rodrigues 93124, Theo Menezes 96145, Renan Ferreira 96168

Universidade de Aveiro

1. Abstract

The Pedagogical Dossier is the essential basis of a Curricular Unit, being the official document that mainly both students and teachers will consult in order to be able to organize themselves in the preparation of teaching/learning the UC. In addition, it also has an outstanding importance in the temporal organization of didactic concepts, indicating the date on which certain contents were taught, allowing later to be consulted in order to clarify the time when the DPUC was active and which themes were addressed.

Particularly in UA, the development phase and the availability of access and reading of a DPUC for the interested academic elements are processes that are not completely cemented, so there is more than one platform where teachers can build a Pedagogical Dossier, and there is a separation between the official DPUC and the dossier that is presented to students on eLearning, with teacher often rewriting a separate document to make the DPUC available to students.

Taking into account the key points mentioned above, this project was proposed to us at the beginning of the semester, with the main objectives of making the UA's DPUC the official and unique document of each curricular unit, that is, the only document that collects all the information of a UC, and also create a main platform for building, developing and maintaining each DPUC.

This report will demonstrate the entire evolution process of this platform, explaining all the steps taken for each phase of the project, the main problems found that impacted this evolution and also the solutions discovered for each of them. We will also address the technologies used throughout the development for the different areas of work, these being MySQL for the database and Docker to present the software separate in containers, Spring Boot for the front-end development and template, Git, in particular GitHub, being the repository of the project and its important documents and files, Elastic Search for the search engine and Google Cloud for the final deployment of the platform.

2. Resumo

O Dossier Pedagógico é a base essencial de uma Unidade Curricular, sendo o documento oficial que, principalmente, tanto alunos como professores irão consultar de modo a se poderem organizar na preparação da lecionação/aprendizagem da UC. Além disto, tem também destacada importância na organização temporal dos conceitos didáticos, sendo que indica a data em que determinados conteúdos foram lecionados, permitindo a, mais tarde, poder ser consultado com o intuito de esclarecer a altura em que o DPUC esteve ativo e quais temas foram abordados.

Particularmente, na Universidade de Aveiro, a fase de construção e a disponibilidade de acesso e leitura de um DPUC para os elementos académicos interessados são processos que não estão completamente cimentados, pelo que existe mais que uma plataforma onde os docentes podem construir um Dossier Pedagógico e há uma separação entre o DPUC oficial e o dossier que é apresentado aos alunos na plataforma elearning, sendo que os professores muitas vezes reescrevem um documento separado para disponibilizar o DPUC para os alunos.

Tendo em conta os pontos fulcrais referidos anteriormente, foi nos proposto este projeto no início do semestre, com os objetivos principais de tornar o DPUC da UA no documento oficial e único de cada unidade curricular, ou seja, o único documento que recolhe toda a informação de uma UC, e também criar uma plataforma principal de construção, desenvolvimento e manutenção de cada DPUC.

Neste relatório irá ser demonstrado todo o processo de evolução desta plataforma, explicando todos os passos realizados para cada fase do projeto, quais os principais problemas encontrados que tiveram impacto nesta evolução e também as soluções descobertas para cada um deles. Iremos também abordar as tecnologias utilizadas ao longo do desenvolvimento para as diferentes áreas, sendo estas MySQL para a base de dados e Docker para apresentar o software separado em containers, SpringBoot para o desenvolvimento do backend, React para o desenvolvimento e template front-end, Git, em particular GitHub, sendo o repositório do projeto e dos seus documentos e ficheiros importantes, Elasticsearch para o mecanismo de pesquisa e Google Cloud para o deploy final da plataforma.

3. Acknowledgements

Gostaríamos principalmente de agradecer ao nosso professor e orientador de projeto José Neto Vieira, que sem a sua grande ajuda e disponibilidade para ultrapassar diversos obstáculos e dificuldades que apareceram pelo caminho não seria possível realizar este projeto.

Um especial agradecimento também ao professor Tomás Oliveira e Silva que também esteve presente em todas as fases da sua evolução.

Queremos também agradecer a todos os professores, colaboradores e funcionários que ajudaram no contínuo acompanhamento da visão global do projeto, assim como no desenvolvimento das várias etapas deste.

1. Abstract.....	2
2. Resumo	3
3. Acknowledgements.....	4
Dossie Pedagógico Único na UA.....	8
4. Introdução	8
4.1. Contexto.....	8
4.2. Motivação	8
4.3. Objetivos	8
5. Métodos.....	8
5.1. Problemas encontrados.....	9
6. State of the Art.....	10
6.1. Trabalho Relacionado.....	10
6.2. Tecnologias.....	14
1.1. React.....	14
1.2. Spring Boot.....	14
1.3. MySQL.....	15
1.4. Microsoft Azure.....	15
1.5. Docker	15
Git	16
Elasticsearch	16
Kibana.....	16
7. Requisitos e do Sistema.....	16
Recolha de Informação	16
Contexto.....	17
Stakeholders.....	18
7.1. Atores.....	18
Diretor de Unidade Orgânica (DUO).....	18
Docente Responsável (DR).....	18

Serviços de Gestão Académica (SGA)	18
7.2. Casos de uso	19
Criação de uma nova DPUC	19
Edição de uma DPUC	20
7.3. Ciclos de Vida.....	21
7.4. Requisitos Funcionais.....	24
7.5. Requisitos Não-funcionais.....	25
7.6. Modelo Tecnológico	26
8. Implementação.....	26
8.1. Base de dados	26
8.2. Backend.....	30
Tecnologias e Dependências	30
Estrutura.....	30
Conexão à base de dados	32
APIs disponíveis	32
8.3. Frontend.....	33
Requisitos	33
Tecnologias.....	35
Componentes	36
Problemas e Desafios.....	50
Testes e Comentários.....	52
8.4. Autenticação.....	53
Backend	53
Frontend.....	56
8.5. Pesquisa.....	57
9. Conclusões e Trabalho Futuro.....	59
9.1. Resultados Práticos.....	59
Criação.....	60

DOSSIE PEDAGOGICO UNICO NA UA	7
Edição	60
Outras funcionalidades	61
10. Conclusões e execução do projeto	61
Execução do projeto	61
Localmente	61
Azure Cloud.....	62
11. Trabalho Futuro.....	62
12. Bibliografia.....	63
13. Anexos	64
13.1. Anexo I.....	64

Dossie Pedagógico Único na UA

4. Introdução**4.1. Contexto**

A criação e elaboração do Dossier Pedagógico de uma Unidade Curricular, assim como a sua atualização periódica, são aspetos vitais para se obter a base e apresentação sólida de uma UC. No entanto, devido a vários sistemas a que este processo está associado no decorrer dos dias de hoje não permite uma centralização dos dados o que pode tornar pouco transparente a forma com que este processo decorre dentro dos sistemas atuais.

4.2. Motivação

Com este intuito, foi-nos proposta a construção e o desenvolvimento de uma plataforma única de desenvolvimento de DPUCs. Tentando assim mitigar alguns problemas da descentralização do processo de criação e edição de um DPUC e mostrar possível, através de uma prova de conceito, um sistema que permite aos docentes, diretores de unidade orgânica e serviços de gestão académica gerir as suas responsabilidades de uma forma facilitada.

4.3. Objetivos

Temos assim como objetivos da plataforma não só permitir todo o processo de criação, manutenção e sustentabilidade de um DPUC num só lugar, ao contrário do sistema atual em que existem 3 pontos de acesso (acesso, portal e paco), mas também ser um ponto de acesso para os vários atores intervenientes neste mesmo desenvolvimento, de modo a, no final, existir apenas um DPUC único, que será o documento oficial consultado tanto pelos docentes de uma determinada unidade curricular como pelos alunos que a frequentam.

5. Métodos

No decorrer deste projeto, e de forma a responder as necessidades das entidades envolvidas no processo de criação e edição de um DPUC, tivemos reuniões frequentes com o prof. José Viera e Tomás Oliveira e Silva no sentido de levantar estes requisitos. Estas reuniões ocorriam na terça-feira às 17h e duravam cerca de duas horas. Muitas das vezes

eram acompanhados de elementos dos STIC como também outros professores e elementos da reitoria no sentido de limar e ter diferentes perspetivas sobre todo este processo.

A nível interno, isto é organização interna entre grupo, comunicamos tanto pelo Discord bem como pelo Whatsapp, com ajuda do orientadores tínhamos sempre tarefas atribuídas semanalmente a todas as pessoas, o que nos ajudou bastante na gestão de prioridades do backlog, para este efeito utilizamos o Jira.

A nível da organização da equipa pelos principais módulos do projeto, esta foi a divisão:

Front-end

- Afonso Rodrigues
- Jorge Oliveira

Back-end

- Afonso Boto
- Tomás Candeias

Base de Dados

- Afonso Boto
- Tomás Candeias
- Theo Menezes

Search tool

- Renan Ferreira

5.1. Problemas encontrados

O principal problema encontrado foi durante as fases iniciais do projeto, pois não estávamos familiarizados com todos os processos que permeiam a criação de uma nova Unidade Curricular e seu Dossiê Pedagógico.

Este fator fez com que tivéssemos que modificar nosso calendário para nos enquadrarmos nessa nova realidade. A nossa solução foi ter reuniões semanais com nosso orientador José Vieira para conseguir fazer com que a recolha dos requisitos ocorresse de forma mais realista, abstraindo processos burocráticos para simplificar a aplicação.

Além disso, como será aprofundado posteriormente, tivemos problemas no Frontend da aplicação e no módulo de busca. Em relação ao Frontend o problema foi utilizar os componentes do PACO 2. A falta de conhecimento em sistemas de busca por texto avançadas, e de processamento das linguagens naturais dificultou no início a elaboração do módulo de busca.

6. State of the Art

6.1. Trabalho Relacionado

A nossa aplicação baseou-se no processo de criação de um DPUC vigente na Universidade, no qual diversos atores fazem parte.

Além disso, utilizamos como base os modelos dos diferentes DPUCs existentes nos pontos de acesso, além do modelo da A3ES, para gerar a especificação dos campos necessários para se gerar um Dossiê Pedagógico, tendo em vista unificar todos os modelos para termos um documento único que pode ser usado em diferentes ocasiões.

O atual processo burocrático relacionado a criação de uma nova UC e seu respetivo DPUC é o seguinte:

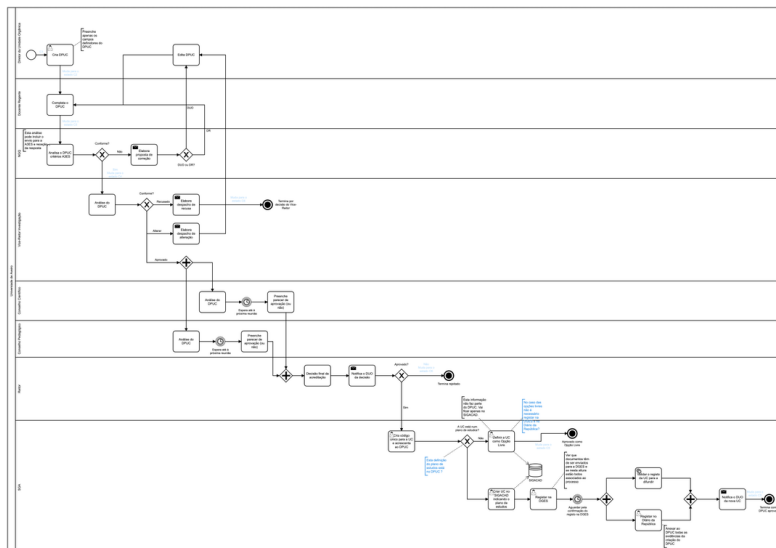


Figura 1 - processo administrativo de criação de uma DPUC

Após a criação do primeiro DPUC, nos próximos anos em que a Unidade Curricular for lecionada, novas edições serão criadas a partir do seguinte processo:

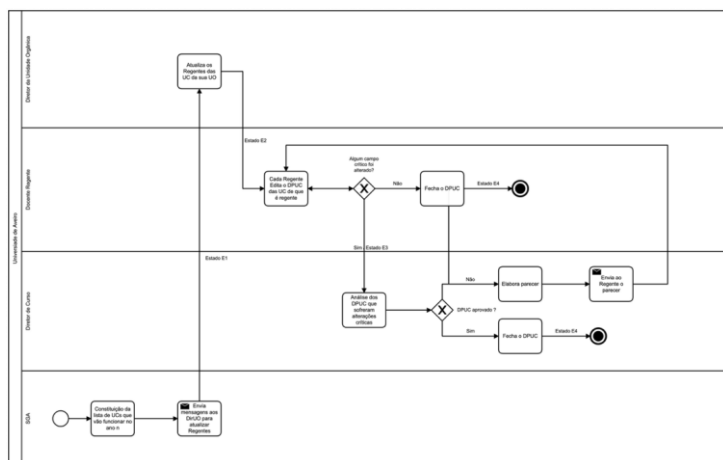


Figura 2 - processo administrativo de edição de um DPUC

Como dito, atualmente os DPUCs ficam armazenados em diferentes lugares, sem conformidade de campos, como pode ser observado nas seguintes imagens:

universidade
de aveiro

[acesso.ua.pt](#)

[jvieira@ua.pt](#) fechar sessão

cursos
unidades curriculares

Sistemas Multimédia (40432)

Responsável: Prof. Doutor José Manuel Neto Vieira (jvieira@ua.pt)
 Área científica: Informática / Sistemas de Informação (ISI)
 Última alteração: 15 de julho de 2018

Cursos associados

Licenciatura em Tecnologias e Sistemas de Informação

[Importar informação do PACO](#)

[voltar](#)

editar informação
 versão inglesa

 objetivos
 conteúdos
 requisitos
 metodologia
 competências
 avaliação
 bibliografia

 carga horária
 página pública

© Universidade de Aveiro 2006-2022
desenvolvimento: STIC | service desk | perguntas frequentes

Figura 3 - Sistema atual acesso


Figura acesso

Sistemas Multimédia

objetivos

Descrições de intenção do processo de ensino, indicando as áreas específicas que o docente pretende abordar numa unidade de aprendizagem (ECTS User's Guide, 2009).


Portuguese:



- Entender a relevância dos sistemas multimédia.
- Entender a necessidade do uso de técnicas para a representação eficiente de informação multimédia.
- Conhecer as principais tecnologias e normas para a representação eficiente e processamento de informação multimédia.
- Conhecer os principais fatores relacionados com o desempenho dos sistemas multimédia.

ul

inglês:



- To understand the relevance of the multimedia systems.
- To understand the need for the use of efficient techniques for representing multimedia information.
- Know the key technologies and standards for the efficient representation and processing of multimedia information
- Know the main factors related to the performance of multimedia systems.

ul

gravar

voltar

Figura 4 - formulário de edição acesso

Lista de UCs
Lista de UCs (PACO)

Unidades Curriculares

ano: 2018 - 2019
 semestre: Todos
 perfil: Docente
 estado: Todos
 procurar

Código	Nome	Período	Perfil	Estado
47134	Introdução à Análise e Processamento de Sinal	2º sem. 2018/2019	Docente	
47644	Introdução à Engenharia Biomédica	1º sem. 2018/2019	Docente	
41511	Introdução aos Sistemas de Radar e Sonar	1º sem. 2018/2019	Docente	

©2013 Universidade de Aveiro

Figura 5 - lista de DPUC no acesso

Sistemas Multimédia

Informação base

código: 40432
 nome: Sistemas Multimédia
 responsável: José Manuel Neto Vieira (jnvieira@ua.pt)
 área científica: Informática / Sistemas de Informação (I/SI) (CC)
 créditos ects: 4

nota ⁽¹⁾:

bibliografia base ⁽²⁾:

está a usar 0 de 400 caracteres.

tipo de disciplina:

Comum

créditos parciais ⁽⁴⁾:

0

componente ⁽³⁾:

não aplicável

gravar informação

campo obrigatório

⁽¹⁾ Para referência de edição (por exemplo, para distinguir unidades curriculares com nomes semelhantes).

⁽²⁾ Formato: Título / Autor / Ano

⁽³⁾ Texto descritivo opcional para os grupos de opção.

⁽⁴⁾ Apenas para unidades curriculares pertencentes a projectos temáticos.

⁽⁵⁾ Apenas para unidades curriculares associadas a cursos Técnicos Superiores Profissionais.

voltar

Figura 6 - formulário de edição PACO

unidade curricular

Introdução à Análise e Processamento de Sinal (47134)

período

2º Semestre 2018/2019

estado

fechado

introdução

caraterização geral

O Dossie Pedagógico de Unidade Curricular (DPUC) é um elemento fundamental na gestão do processo académico, coligindo informação essencial para o bom funcionamento da atividade letiva e contribuindo para a melhoria da qualidade do processo de ensino e aprendizagem.

Este novo modelo de DPUC permite incorporar a informação solicitada no REUA e facilitar o preenchimento dos relatórios de autoavaliação, completando parte da informação pedida pela ASES - Agência de Avaliação e Acreditação do Ensino Superior.

O presente documento será construído em 3 momentos distintos. No primeiro momento, a decorrer no período definido no REUA e refletido no Calendário do Ciclo Académico, serão preenchidos os campos relativos às características gerais da unidade curricular, modo de lecionação, objetivos/conteúdos, metodologias, avaliação e análise da coerência entre os conteúdos programáticos e os objetivos da unidade curricular, e entre as metodologias de ensino e de avaliação e os objetivos de aprendizagem da unidade curricular.

O segundo momento, a ocorrer no final da época de exames do respetivo semestre, servirá para constituir um repatório de informação (por exemplo os enunciados das provas escritas, guões de aulas práticas, ...).

O terceiro momento consistirá na importação automática de informação do SubSG LUC. Após a conclusão do processo, o docente responsável será informado via email da disponibilização da versão final do DPUC.

Os campos do DPUC serão preenchidos em português e inglês. Para utilizadores da versão em Língua Portuguesa, a interface será sempre em Língua Portuguesa. Para utilizadores da versão em Língua Inglesa, que deverão ativar a versão inglesa através do botão de topo direito (EN), a interface será sempre em Língua Inglesa. A versão final do DPUC será bilingue.

importar dados do Acesso

importar dados do DP anterior

fechar dossiê

versão portuguesa (pdf)

versão inglesa (pdf)

	pt	en
objetivos de aprendizagem	✓	✓
conteúdos programáticos	✓	✓
articulação das atividades	✓	✓
metodologias de ensino e de avaliação	✓	✓
aprendizagem ativa	✓	✓
cálculo da classificação final	✓	✓
requeritos	✓	✓
bibliografia principal	✓	✓
coerência conteúdos / objetivos	✓	✓
coerência metodologias / objetivos	✓	✓
modalidade de lecionação	✓	✓
língua(s) de lecionação	✓	✓
horas de contacto	✓	✓

©2013 Universidade de Aveiro

Figura 7 - vista de um DPUC PACO

6.2. Tecnologias

1.1. React

React.JS (React.JS, s.d.) é uma biblioteca em *JavaScript* para desenvolver interfaces de utilizador através de páginas *web*. Nesta biblioteca existe o uso de *HTML*, *CSS* e *JavaScript* para criar componentes com funcionalidades e visuais diferentes que, em conjunto, constituem uma página completa. Foi feita a escolha de *React.JS* para o projeto dado que as páginas *web* do PACO2 estão a ser desenvolvidas nesta mesma biblioteca e é também uma oportunidade para expandir os conhecimentos de desenvolvimento *web* através de páginas altamente dinâmicas e responsivas com o princípio de aplicação de uma página (*Single Page Application*).

1.2. Spring Boot

Este projeto foi desenvolvido com recurso ao Spring Boot. O Spring Boot é um conjunto de frameworks que permitem a fácil execução e desenvolvimento de aplicações independentes e prontas a correr. Da maneira que está estruturado permite manter o controlo

de versões, e fazendo o uso das ferramentas que são disponibilizadas conseguimos abstrair certas componentes da implementação com recurso a anotações Spring. Neste caso usamos várias frameworks derivados da Spring framework, tais como Spring Security, Spring Data e Spring Web. Tudo adicionado permitiu-nos uma produção rápida e eficaz.

A nossa familiaridade desta ferramenta em projetos anteriores levou com que o Spring Boot fosse uma ferramenta de eleição.

1.3. MySQL

Para a base de dados relacional da plataforma foi utilizado MySQL.

Esta escolha deveu-se principalmente à familiaridade de utilização desta em projetos passados, tanto deste sistema como também da linguagem SQL, mas também por ser um software com alta escalabilidade e flexibilidade. Para além disso, o MySQL possui um excelente desempenho, mesmo com grandes cargas de informação, o que para o nosso projeto é o caso, e tem uma fácil integração com o Docker.

1.4. Microsoft Azure

Para fazer o deploy da aplicação utilizamos a Microsoft Azure.

Esta escolha ocorreu devido à facilidade que encontramos em utilizar a plataforma, principalmente devido à apresentação que tivemos em um dos *workshops*. Além disso, possuímos acesso à funcionalidades pagas através dos créditos fornecidos pela universidade.

O Microsoft Azure permite utilizar máquinas virtuais para configurar e hospedar uma aplicação web, onde pudemos armazenar nosso repositório e correr os dockers.

1.5. Docker

Sendo o docker um software de *Platform as a Service* (PaaS) que usa virtualização ao nível do sistema operativo de forma a disponibilizar o software em containers.

O nosso docker-compose é composto por dois containers: mysqldb (base de dados MySQL) e fetcher-main (API central, com toda a lógica de manipulação de dados e autenticação). Com isto, para garantir um bom funcionamento da aplicação foi adicionado ao container mysqldb um *health check* que vai verificando se o container está a correr, sendo que o fetcher-main depende disto para funcionar.

Foi sem dúvida uma mais valia no trabalho, garantiu-nos ter sempre a base de dados consistente e persistente entre todos os elementos, pois o volume da base de dados era submetido para o repositório no GitHub cada vez que eram feitas alterações relevantes às

tabelas ou à dados de teste, e nos restantes estava no ficheiro `.gitignore` para as manipulações locais não afeteram o “estado de vida” dos dados da base de dados.

É de notar que, apesar de não estar integrado com o nosso projeto, estão em comentário no `docker-compose` os containers para o `elastic search`, `search API` e `kibana` para futuros grupos poderem logo de imediato dar uso.

Git

O Git é uma ferramenta de linha de comando de controle de versão muito popular para gerir o código de um programa, especialmente quando o mesmo é implementado em equipas. Para garantir que a equipa pudesse trabalhar em contexto relativamente independente e pudesse garantir que qualquer novo tipo de código não prejudicasse o já implementado e funcional, escolheu-se usar a mesma para uma boa gestão do desenvolvimento em grupo. Além disso usamos o GitHub, uma ferramenta que permite guardar o código remotamente para poder ser facilmente compartilhado, além de permitir o uso do Git em ambiente gráfico.

Elasticsearch

Elasticsearch é uma ferramenta de pesquisa, de formato distribuído e replicado, em nós que se agrupam em grupos chamados de *clusters*. A ferramenta é conhecida por oferecer pesquisa em texto rápidas, eficientes e relevantes. Os dados são guardados em formato de documento *JSON*, em índices.

Kibana

Relacionado ao *Elasticsearch*, *Kibana* é uma plataforma que permite administrar os nós e *clusters*, obtendo também métricas relacionadas aos dados dos mesmos, além de outras métricas como número de requisições e média de atualização dos dados.

7. Requisitos e do Sistema

Recolha de Informação

O projeto surgiu como uma necessidade da Universidade de Aveiro de centralizar as informações referentes ao dossier pedagógico das Unidades curriculares com vista a facilitar a vida aos docentes e aos estudantes.

A ideia do mesmo veio do Pró-reitor José Vieira, que também atuou como orientador para o projeto, e foi ele que nos transmitiu grande parte da problemática atual dos variados órgãos da Universidade referente a gestão deste tipo de dado, além claro, dos estudantes, que

muitas vezes não conseguem obter toda informação relevante sobre suas próprias cadeiras em um único ponto.

Através de reuniões regulares com o orientador e algumas reuniões com membros dos STIC, conseguimos recolher informações sobre como funciona os processos de criação e edição de um Dossier Pedagógico. A partir dessas reuniões com os mesmos, conseguimos obter os principais requisitos gerais de como a nossa aplicação deveria funcionar.

Contexto

O contexto inicial que nos foi apresentado foi da dificuldade dos estudantes de encontrar as informações inerentes as unidades curriculares que os mesmos tem. Estudantes não conseguem obter o Dossier Pedagógico de uma Unidade Curricular (DPUC), em um único ponto, além claro, da dificuldade dos profissionais da Universidade de Aveiro em gerir esta informação, que está espalhada por diferentes componentes. Daí surgiu a ideia do projeto, com os seguinte requisitos iniciais.

1. Princípio “apenas uma vez”. Ou seja, a mesma informação só deve ser pedida aos docentes uma vez.

2. Princípio de DPUC mestre. Uma vez criado e validado de forma oficial o DPUC, este deve ser armazenado num sistema informático que deve servir de mestre de todos os outros que apenas poderão ler informação deste sistema.

3. Princípio da Universalidade. A informação contida neste novo dossier pedagógico foi pensada de modo a ser o mais abrangente possível. Tem todos os campos necessários à elaboração das fichas de UC para os processos da A3ES e de outras agências de certificação.

4. Princípio de contrato. O DPUC deve ser encarado como um contrato entre o docente responsável e os alunos que vão frequentar a UC.

Porém, com o início da elaboração do projeto, começamos a perceber que faltava entender mais aspectos da gestão de um Dossier Pedagógico, o que envolvia processos altamente complexos. Assim, juntamente com o Professor José Vieira, aprendemos como funciona o processo de criação e edição de um Dossier Pedagógico, principalmente no que se refere as entidades responsáveis pela sua aprovação: A3ES, o Conselho Pedagógico, o Conselho Científico, e o Reitor.

A partir disso definimos dois casos de uso principais: A criação de um dossier pedagógico, e a edição de um dossier pedagógico. Além desses dois requisitos funcionais, o professor José Vieira sugeriu alguns outros relacionados a visualização dessas informações, a partir de conversas com ele, e o apoio do Professor Sérgio Matos, viemos com mais um requisito: Ser capaz de buscar resultados, nesse caso, unidades curriculares, relevantes para um utilizador universal, especialmente aos profissionais da Reitoria, como dito pelo Professor José Vieira.

Stakeholders

Os Principais Stakeholders do nosso serviço são:

- Diretores de Unidades Orgânicas
- Professores Regentes
- Profissionais dos Serviços de Gestão Académica (SGA) da Universidade de Aveiro
- Estudantes
- Utilizadores universais que visitam a página da Universidade

7.1. Atores

Os principais atores da aplicação são:

Diretor de Unidade Orgânica (DUO)

Responsável por iniciar o processo de criação de um novo UC e definição de docente responsável.

Docente Responsável (DR)

Responsável pelo preenchimento de todas as informações do DPUC.

Serviços de Gestão Académica (SGA)

Valida a passagem de toda a informação. Também fica responsável por requisitar a aprovação do DPUC pelas diferentes entidades com este privilégio.

O processo administrativo de Criação e Edição de um DPUC envolve outros autores que não são levados em conta na aplicação, que são as entidades responsáveis pela aprovação do mesmo: O AE3s, o Núcleo de gestão de Qualidade, o conselho Pedagógico, o conselho

Científico, e o Reitor. Os seus respectivos pareceres são gerenciados pelos serviços de gestão acadêmica.

7.2. Casos de uso

Como referido anteriormente, os principais casos de uso da aplicação são os seguintes:

Criação de uma nova DPUC

O diagrama abaixo mostra como os diferentes atores da aplicação possuem diferentes funcionalidades, de acordo com o processo de criação de um novo DPUC. Além disso, o mesmo apresenta as mudanças de estado da mesma durante o processo.

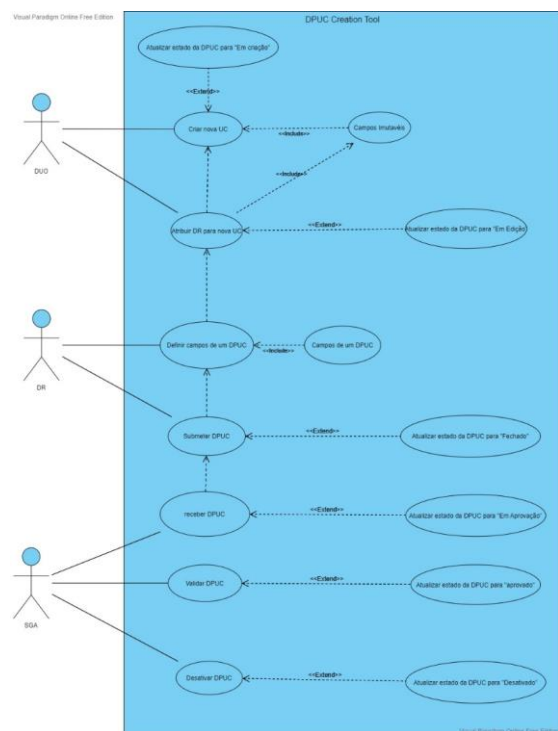


Figura 8 - caso de uso de criação de um DPUC

Edição de uma DPUC

O diagrama, em semelhança ao processo de criação, mostra como os diferentes atores participam do processo de edição de um DPUC, a partir do ponto de vista da aplicação.

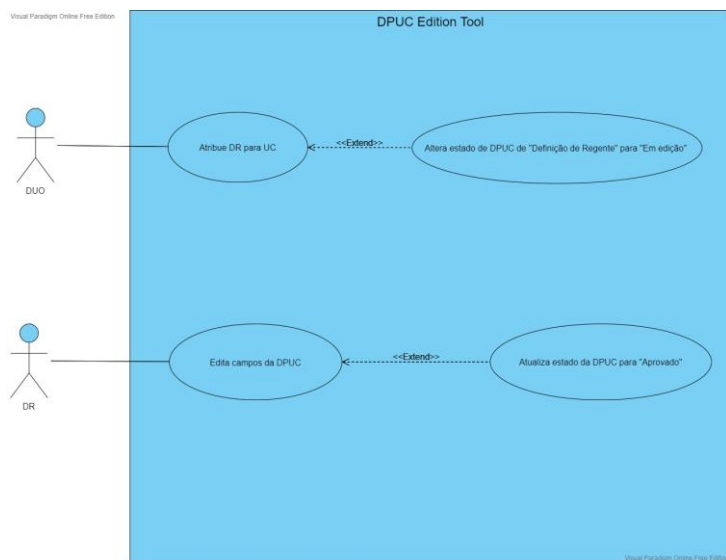


Figura 9 - caso de uso de edição

Buscas por UCs e/ou DRs

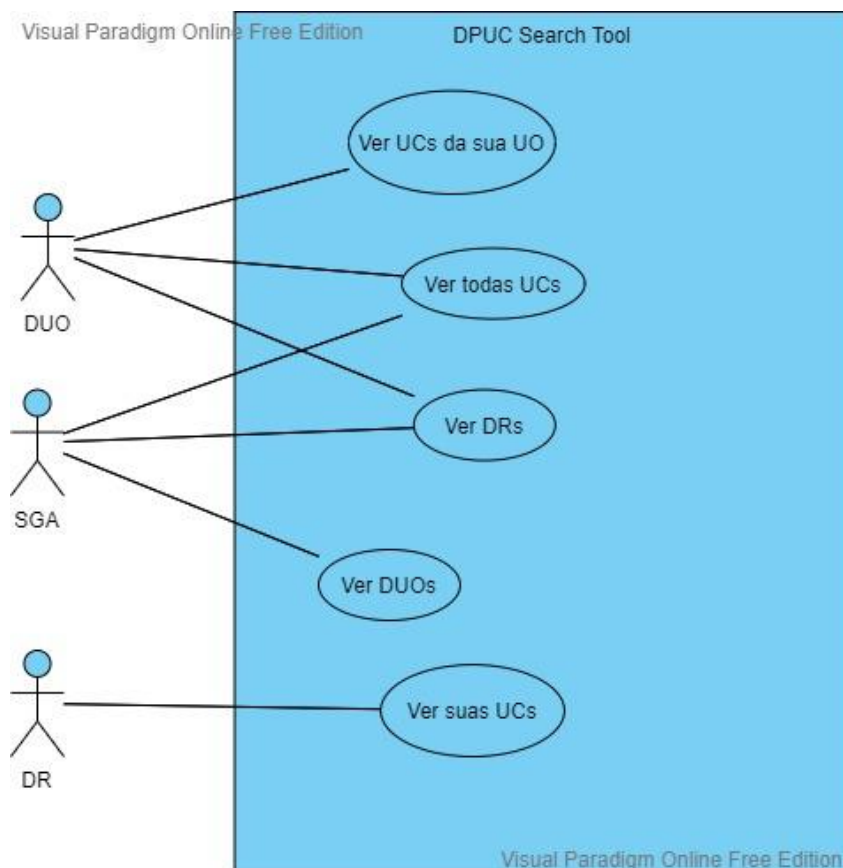
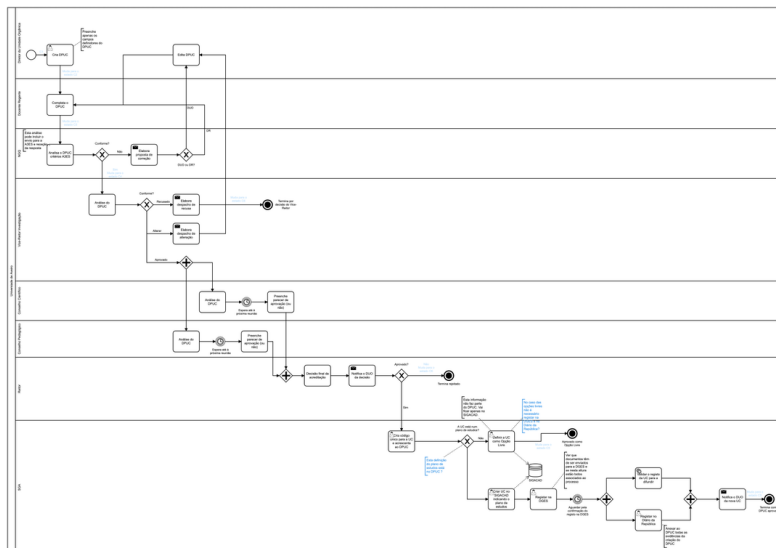


Figura 10 - caso de uso da search tool

7.3. Ciclos de Vida

Ciclo de vida da Criação de uma DPUC

O ciclo de criação de uma DPUC, no contexto administrativo, é colocado abaixo no diagrama do tipo *Business Process Model and Notation* (BPMN).



Para facilitar a implementação na nossa aplicação deste ciclo, foram abstraídos alguns elementos e o resultado final se encontra abaixo.

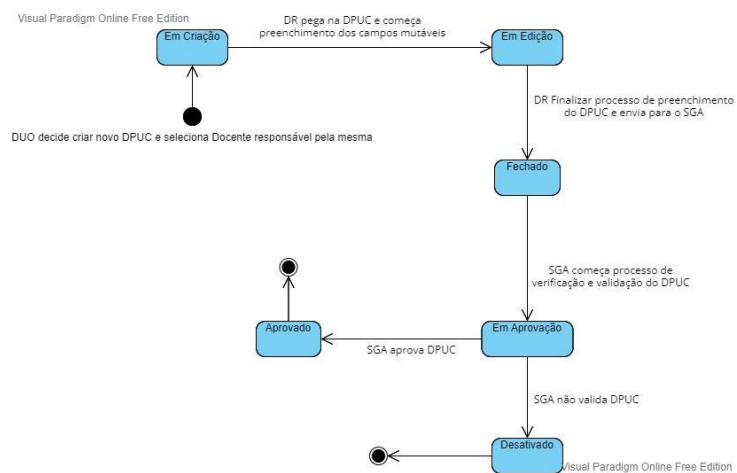


Figura 11 - ciclo de vida de criação de um DPUC

O mesmo exemplifica bem o papel de cada ator no processo e como a nossa aplicação funciona para esse específico contexto.

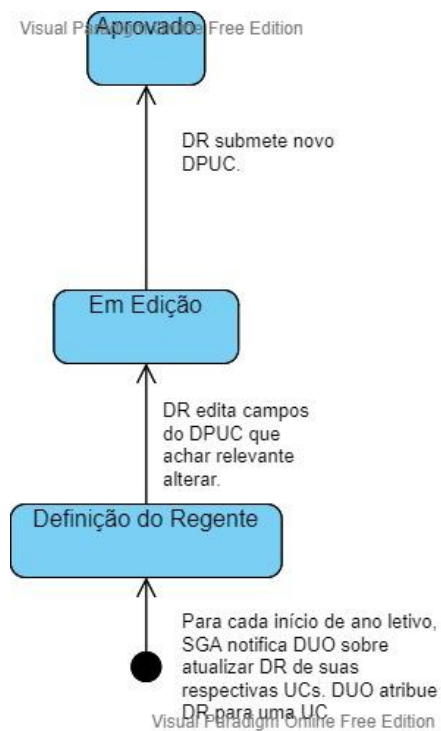


Figura 12 - Ciclo de vida de edição de um DPUC

7.4. Requisitos Funcionais

- Requisitos funcionais DUO:
 - o Criar uma nova UC (passar DPUC para o estado 1);
 - o Submeter primeira versão de DPUC (passar para o estado 2-em edição);
 - o Associar um regente a uma DPUC
- Requisitos funcionais DR:
 - o Fazer edição de um DPUC a si associado;
 - o Guardar atual edição do DPUC que está a editar;

- o Submeter a versão final de um DPUC (passar para o estado 3-fechado);
- Requisitos funcionais SGA:
 - o Editar um DPUC;
 - o Submeter um DPUC para aprovação (passar para o estado 4-em aprovação);
 - o Aprovar um DPUC (passar para o estado 5-aprovado);
 - o Desativar um DPUC (passar para o estado 6-desativado);
 - o Pesquisa de docentes;
 - o Mudar estado de um DPUC para ‘Em Criação(1)’ e ‘Em Edição(2)’;

7.5. Requisitos Não-funcionais

- Usabilidade: a plataforma é fácil de compreender e manusear;
- Compatibilidade: o sistema irá ser compatível e integrável com outros sistemas de administração da UA;
- Segurança: o sistema tem um sistema de Login de acordo com as regras da UA;
- Desempenho: o sistema tem uma boa performance, sendo que é rápido a responder a cada pedido ou alteração feita pelo utilizador;
- Manutenção: sendo um sistema baseado em frameworks utilizados em outros sistemas da UA, a plataforma tem uma boa capacidade de manutenção;
- Fiabilidade: O sistema é tolerante a falhas e erros, permanecendo ligado quando existem problemas de serviços externos;

7.6. Modelo Tecnológico

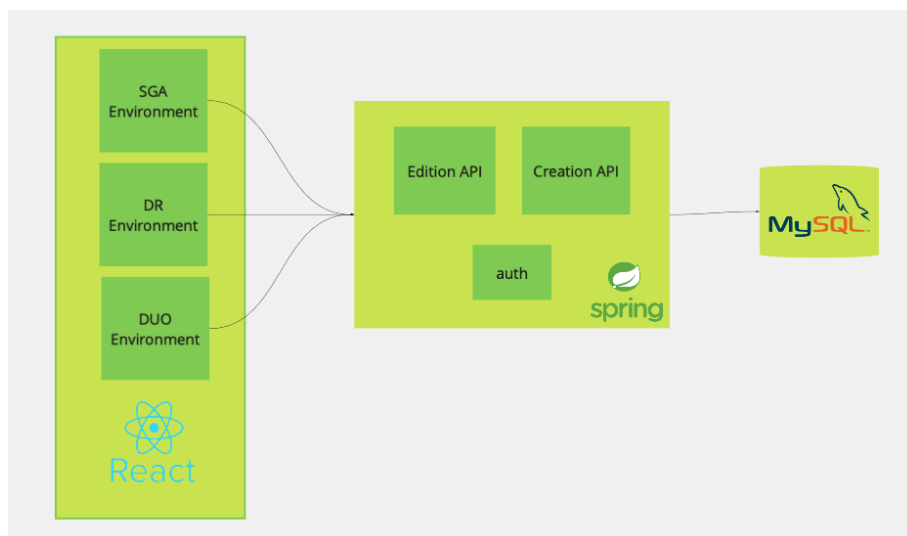


Figura 13 - Modelo Tecnológico

8. Implementação

8.1. Base de dados

Tecnologias

MySQL

Estrutura

Utilizamos MySQL para a base de dados de nosso projeto pois precisávamos de uma solução relacional e que pudesse ser facilmente usada com dockers. Durante o desenvolvimento do projeto a nossa estrutura mudou algumas vezes conforme notávamos a necessidade de alterações para cumprir requisitos.

Nossa primeira versão da base de dados estava estruturada conforme o seguinte diagrama:

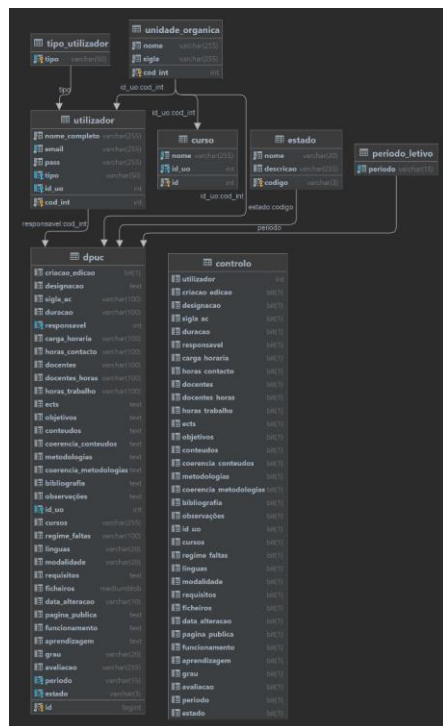


Figura 14 Diagrama de base de dados versão primordial

Nesta versão, a base de dados possuía as seguintes tabelas:

- *tipo_utilizador*: usada para representar o papel do utilizador na aplicação (Diretor de Unidade Orgânica, Docente Responsável ou SGA);
- *unidade_organica*: usada para representar um departamento ou unidade orgânica, possui um nome, sigla e um código interno;
- *curso*: usada para representar um curso, possui um nome, código interno e está relacionada a uma Unidade Orgânica;
- *estado*: usada para representar em qual estado se encontra um DPUC, possui um nome, descrição e código interno;
- *periodo_letivo*: usada para descrever a qual semestre um DPUC pertence;
- *utilizador*: usada para fazer a gestão de utilizadores, possui nome completo, email, palavra-passe, uma chave estrangeira para relacionar ao tipo e outra para a UO, além de um código interno;

- *dpuc*: usada para armazenar os DPUCs, possui todos os campos recolhidos durante a fase de requisitos e depende de outras 4 entidades: *responsavel*, para associar um Docente Responsável; *uo_id*, para associar a uma UO; *periodo*, para associar a um período letivo; e *estado*, para associar a um estado;
- *controlo*: usada para gerir as permissões de alterações de cada campo entre os diferentes tipos de utilizador.

Após algumas alterações por conta de mudanças de requisitos chegamos à seguinte estrutura:

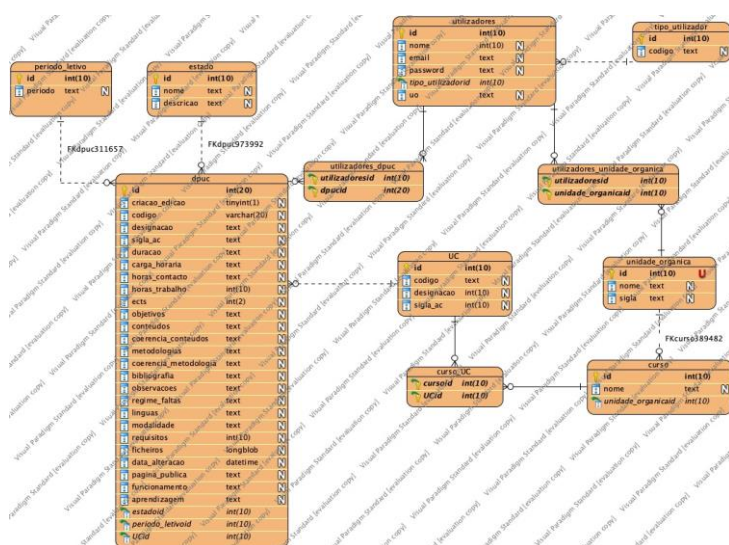


Figura 15 Diagrama de base de dados intermédio

As principais mudanças feitas foram:

- Redução do número de campos do DPUC;
- Introdução da tabela *utilizadores_dpuc*, que associa um docente a um DPUC;
- Introdução do conceito de UC: campos imutáveis de um DPUC, como nome, código e área científica;
- Associação de uma UC a um curso por meio da tabela *curso_uc*;
- Criação da tabela *utilizadores_unidade_organica*, para definir um DUO a cada Unidade Orgânica;
- Remoção da tabela *controlo*, após esse momento todas as definições de permissões são geradas diretamente do *frontend*. Fizemos esta mudança por termos escolhido

portanto podem ocorrer algumas associações estranhas, como uma Unidade Curricular estar relacionada a uma UO diferente da esperada.

8.2. Backend

Tecnologias e Dependências

Spring Boot

Maven

Json Web Token

Spring Data JDBC

Estrutura

A base deste projeto está feita usando a Spring Boot Framework. Neste caso a estrutura dos packages e a maneira com que estes interagem uns com os outros seguem a ferramenta de gestão e compreensão Maven (referida anteriormente). Usamos o pom.xml, obrigatório em projetos Maven, para definir dependências e plugins uteis durante o projeto, bem como a especificação das suas versões, com isto mantemos a coerência das versões conseguindo assim que este projeto seja replicado.

Posto isto, definimos então na pasta *src* deste projeto todas as classes essenciais para o desenvolvimento do projeto sendo elas em implementação ou para testes.

Focando agora na estrutura da implementação do *fetcher-main*, este está dividido em 4 packages centrais: config, controller, entity e service.

Config

Neste package podemos encontrar as classes que auxiliam a autenticação. Estas classes permitem criar JWT (JWT (Json Web Tokens) que vão acompanhar toda a sessão de um utilizador e também de verificar que essa autenticação foi feita.

Controller

Neste package é definida a “fronteira” desta aplicação. Aqui permitimos que componentes externas deste programa possam interagir com o mesmo, seja em extrair ou adicionar informação. Neste package estão divididos entre 3 controladores diferentes: MainController, EditionController e LoginController.

MainController

Aqui são definidos os endpoints que permitem acesso direto a informação bem como a gestão do ciclo de criação das DPUCs.

Edition Controller

Neste controlador é onde é exposta APIs relacionadas com o processo de edição.

Login Controller

Por último temos ainda o controlador do Login, onde é feita o acesso da autenticação e autorização do nosso projeto, é aqui que o frontend vai garantir que um utilizador existe e é quem ele diz ser.

Entity

Neste package é onde está toda a lógica do negócio, são todas elas mapeadas pela base de dados. Assim conseguimos manipular diretamente os dados fazendo operações sobre as entidades. Sendo elas Área Científica, que contém sigla e designação, Curso, com nome e referencias para a sua unidade orgânica e para o utilizador. Temos ainda a entidade DPUC, que alberga 15 campos, estes referentes a documentos disponibilizados pelo Prof. Jose Vieira.

Temos UC, com código, designação, a sigla da Área Científica, ECTS e ainda a referência para a unidade orgânica. Ainda temos Unidade Orgânica com nome sigla e o referente diretor de unidade orgânica. Estado, com nome e descrição, Período Letivo com o seu nome. Por fim temos utilizadores com nome, número mecanográfico, email, password e o seu tipo de utilizador que define o seu papel no sistema.

Service

Neste package é onde é feita a manipulação direta com os dados através das entidades, que com acesso à base de dados permitem manter a consistência dentro do fetcher-main deste modo o controlador chama estes serviços permitindo assim a outras componentes do projeto, mas fora deste programa fazer uso desses controladores. Estão definidos 4 serviços principais: Creation Manipulation Service, Creation Service, Edition Service e Login Service. O Creation Manipulation Service serve para escrita, sendo que cria e altera dados referentes ao processo de criação. O Creation Service que serve para leitura, sendo lê os dados e os disponibiliza para o controlador. O Edition Service refere-se ao processo de edição. E ainda o Login Service que trata de dar ao config a existência e uso correto de credenciais para acesso externo.

Conexão à base de dados

A conexão à base de dados MySQL foi feita com a ajuda do Docker e definida no container fetcher-main no SPRING_APPLICATION_JSON.

Em vários trabalhos realizados na universidade fomos familiarizado com o Spring Data JPA, porém com vista à continuidade do trabalho e a fácil migração para outras frameworks acabamos por utilizar o Spring Data JDBC. Este têm a mesma finalidade que o outro, mas o nível de abstração não é tão grande, podendo assim uma muito mais fácil manipulação dos dados através de queries diretas à base de dados.

APIs disponíveis

Os endpoints foram documentados com recurso ao Swagger que é uma plataforma que ajuda o desenvolvimento de API's, neste caso a documentar.

Posto isto recorro ao Anexo I para consulta das APIs.

8.3. Frontend

Requisitos

Para satisfazer os processos de Criação e Edição de um DPUC e facilitar navegação aos utilizadores finais foram estabelecidos alguns requisitos adaptados dos requisitos gerais do projeto para as páginas *web*:

Requisitos Funcionais Gerais

<i>RFG-1</i>	Ver a lista de DPUCs de Unidades Curriculares (UC) ativas.
<i>RFG-2</i>	Filtrar DPUCs por estado tal como nome e código da UC respetiva.
<i>RFG-3</i>	Visualizar os detalhes de um DPUC
<i>RFG-4</i>	Acesso a diferentes edições de DPUC de uma UC.

Requisitos Funcionais para Serviços de Gestão Académica

<i>RFSGA-1</i>	Ver todos os DPUCs de UCs de qualquer Unidade Orgânica.
<i>RFSGA-2</i>	Editar qualquer DPUC aberto.
<i>RFSGA-3</i>	Passar um DPUC do estado “Fechado” (C3) para “Em Aprovação” (C4).
<i>RFSGA-4</i>	Passar um DPUC do estado “Em Aprovação” (C4) para “Aprovado” (C5).
<i>RFSGA-5</i>	Desativar um DPUC (Estado C6).
<i>RFSGA-6</i>	Reabrir um DPUC (Passar para o estado “Em Edição” C2 ou E2).
<i>RFSGA-7</i>	Filtrar DPUC por Unidade Orgânica
<i>RFSGA-8</i>	Abrir um novo processo de edição (Iniciar período letivo, cria DPUCs no estado E1).

Requisitos Funcionais para Diretor de Unidade Orgânica

<i>RFDUO-1</i>	Ver todos os DPUCs de UCs da sua Unidade Orgânica.
<i>RFDUO-2</i>	Criar uma nova UC (Estado C1).
<i>RFDUO-3</i>	Submeter primeira versão de DPUC (Estado C2).
<i>RFDUO-4</i>	Lançar um novo DPUC (Passar para estado E2).
<i>RFDUO-5</i>	Alterar Docente Regente de uma UC.
<i>RFDUO-6</i>	Pesquisar Docentes ativos na Universidade.

Requisitos Funcionais para Docente Responsável

<i>RFDR-1</i>	Ver todos os DPUCs de UCs no qual é responsável.
<i>RFDR-2</i>	Editar DPUCs abertos.
<i>RFDR-3</i>	Submeter a versão final de um DPUC (Passar para estado “Fechado” C3).
<i>RFDR-4</i>	Submeter a versão final de uma nova edição de DPUC (Passar para estado “Aprovado” E4).

Requisitos Não Funcionais

<i>RNF-1</i>	Plataforma fácil de compreender e manusear.
<i>RNF-2</i>	Páginas web compatíveis com diversos browsers em formato desktop.
<i>RNF-3</i>	O servidor <i>web</i> tem um bom desempenho com respostas e pesquisas rápidas.
<i>RNF-4</i>	Desenvolvimento baseado em componentes, com manutenção fácil e rápida.
<i>RNF-5</i>	Uso de componentes já presentes nas páginas do PACO 2.
<i>RNF-6</i>	Cumprimento do <i>Design System</i> das interfaces da Universidade de Aveiro.

UA UI Design System

Para cumprir os requisitos não funcionais RNF-1 e RNF-6 foi estudado o *Design System* da UA (UA Design System, s.d.) com foco especial das *Guidelines* para *Color*, *Content* e *Typography*. Ao implementar as técnicas usadas nas páginas da Universidade de Aveiro as páginas web do nosso projeto já vão parecer familiares aos docentes utilizadores dado que o estilo de letras e cores é igual ao Portal da UA e PACO 2.

Tecnologias

Para desenvolver as páginas *web* tirámos partido de algumas tecnologias e bibliotecas disponíveis para *React.JS*. Estas tecnologias permitiram-nos cumprir os requisitos das páginas *web* tal como facilitar o processo de desenvolvimento em termos de estilo.

PACO2 UI

Os desenvolvedores do PACO 2 disponibilizaram um pacote com componentes de *React* (PACO UI NPM, s.d.) no qual foi feito o uso de alguns dos mesmos nas nossas páginas *web* para manter a identidade da UA na parte visual do projeto.

Axios

Axios (Axios, s.d.) é um cliente HTTP baseado em “promessas” implementado em *JavaScript* que pode ser utilizado em qualquer browser para fazer pedidos a servidores externos. No percurso do projeto foi utilizada este cliente para comunicar com o *Fetcher* e pedir as entidades necessárias (Unidades Orgânicas, Docentes, DPUCs, etc.) e fazer uso dos métodos da API para cumprir os processos de criação e edição de um DPUC.

React Bootstrap

React Bootstrap (React Bootstrap, s.d.) é um *toolkit* para ajuda no desenvolvimento *frontend* que facilita a criação de uma interface altamente responsiva e reduzir um pouco o trabalho na estilização de alguns componentes. Este módulo foi maioritariamente usado para organizar a estrutura visual, alguns componentes visuais também foram adaptados e transformados para transmitir a identidade da UA.

React Router

React Router (React Router, s.d.) é uma biblioteca de roteamento para *React* que nos permite gerir os caminhos, redireccionamentos e a navegação dentro da nossa aplicação *web*.

Componentes

Nesta secção existirá uma descrição e funcionamento de alguns dos componentes gerais dentro da nossa aplicação *web*, há componentes que são dependentes entre si, no entanto dada a facilidade de exportação no *React* estes podiam ser exportados para qualquer outro projeto.

Barras de Navegação

No nosso *website* desenvolvemos diferentes barras de navegação simples para recriar as do PACO 2 dado que não nos disponibilizaram uma versão destes componentes. O que foi implementado na nossa versão tem uma funcionalidade semelhante à da barra de navegação “oficial”. Como uma pequena ajuda, antes do nome é indicado o tipo de ator no utilizador do nosso sistema.



Figura 17 A nossa Implementação da barra de navegação



Figura 18 Barra de Navegação do Paco2

Além de barras de navegação também foi feito um rodapé muito simples baseado nos rodapés do PACO 2.

Figura 19 Rodapé no nosso projeto

Componentes Visuais

Pré-visualização de DPUC – CardDPUC

Para representar os DPUCs numa lista, criámos um componente facilmente utilizável, onde são mostradas algumas informações úteis sobre um DPUC. Este componente sofreu algumas transformações ao longo do projeto:

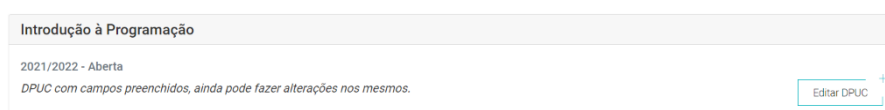


Figura 20 Versão 1



Figura 21 Versão 2



Figura 22 Versão 3 – Final

Como é possível observar, houve uma clara evolução visual para cumprir as normas de *design* da UA e foi adicionada bastante mais informação sobre a UC do DPUC em questão como o Código, Unidade Orgânica, Regente e Edição. É de notas que dependendo dos estados e utilizador, estas pré-visualizações apresentam ações diferentes: Ver, Editar, Alterar Regente e lançar novo DPUC.

Lista de DPUCs – DPUCList

Para obter uma forma de visualizar rapidamente diversos DPUCs foi feito um componente que lista DPUCs disponíveis aos utilizadores com mecanismos de filtragem em pesquisa. A listagem dos DPUCs faz uso do *CardDPUC* descrito anteriormente.

[Iniciar Processo De Edição](#)

Filtrar DPUCs por Unidade Orgânica:
 Todas - Qualquer Unidade Orgânica

Pesquisa de DPUC:
 Pesquisa por nome, código ou docente regente

Filtrar DPUCs por estado:
 Mostrar Todos (1338)

<p>[44535] DLC - CULTURA PORTUGUESA CONTEMPORÂNEA</p> <p>Regente: António Nuno Rosmaninho Rolo</p> <p>Última alteração: 2021-09-01</p> <p>DPUC Aprovado.</p>	<p>Aprovado (5)</p> <p>Edição de 2021/2022</p> <p>Ver DPUC</p>
<p>[40379] DETI - FUNDAMENTOS DE PROGRAMAÇÃO</p> <p>Regente: João Manuel de Oliveira e Silva Rodrigues</p> <p>Última alteração: 2021-09-01</p> <p>DPUC Aprovado.</p>	<p>Aprovado (5)</p> <p>Edição de 2021/2022</p> <p>Ver DPUC</p>
<p>[42709] DAO - ÁLGEBRA LINEAR E GEOMETRIA ANALÍTICA</p> <p>Regente: Maria Raquel Rocha Pinto</p> <p>Última alteração: 2021-09-01</p> <p>DPUC Aprovado.</p>	<p>Aprovado (5)</p> <p>Edição de 2021/2022</p> <p>Ver DPUC</p>

Figura 23 Listagem de DPUCs (Vista de SGA)

Em exclusivo na vista de utilizadores pertencentes aos Serviços de Gestão Académica encontra-se um botão para iniciar o processo de edição (na realidade esta funcionalidade seria automática no sistema dado que só existe um processo de edição por ano/semestre) e também filtragem de DPUCs por Unidades Orgânicas. A filtragem por UO não está presente nos outros tipos de utilizadores já que só aparecerão Dossiers Pedagógicos da sua Unidade Orgânica. Por outro lado, os Diretores de Unidade Orgânica (DUO) possuem uma funcionalidade de criar uma Unidade Curricular.

Dado que o módulo de pesquisa de DPUCs (*search-api*) não estava completo e corretamente funcional, foi efetuada uma pesquisa de DPUCs na própria aplicação *web* para onde qualquer utilizador consegue fazer as pesquisas de UCs pelo nome ou o docente responsável/regente da mesma. Para complementar esta pesquisa também existe uma filtragem de DPUCs por estado e, em exclusivo para os SGA, também é possível filtrar pela Unidade Orgânica das Unidades Curriculares.

Para adicionar um pouco de informação ao utilizador, na filtragem por estado também são indicadas a quantidade de DPUCs para cada estado de forma a ser possível verificar rapidamente se há algum num estado intermédio (“Edição”, “Fechado” ou “Em Aprovação”).

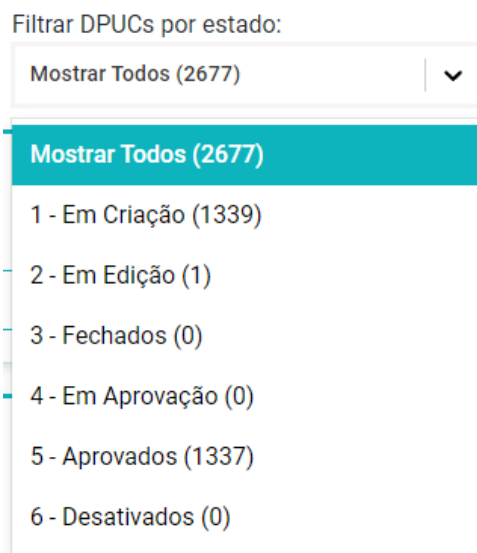


Figura 24 Filtragem de DPUCs por estado

Para facilidade de navegação existe paginação de DPUCs quando existe um elevado número de DPUCs. Desta forma é mostrada uma quantidade reduzida de DPUCs de cada vez o que melhor a experiência do utilizador a nível visual e também cria uma grande melhoria no desempenho da aplicação.

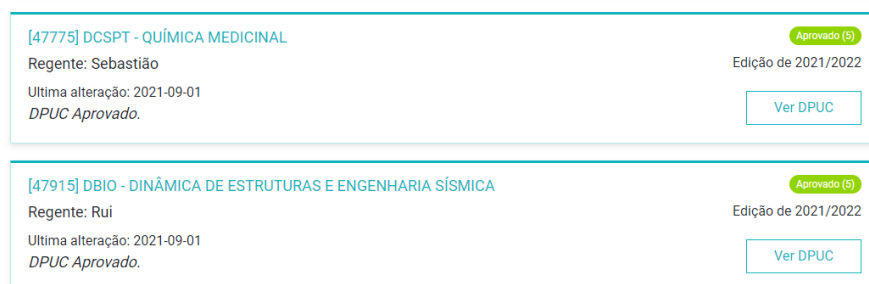


Figura 25 Sistema de Paginação

Edições de DPUC – DPUCVersions

Na visualização de DPUCs é dada uma opção ao utilizador de escolher qual edição de DPUC de uma UC pretende visualizar, este componente pega nos anos letivos dos DPUCs da mesma UC para gerar uma lista simples com as edições disponíveis.

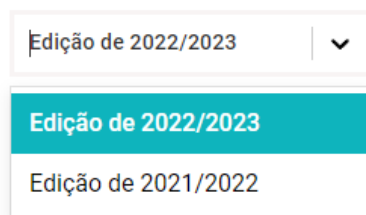


Figura 26 Edições de um DPUC

Selecionador de Docentes – SelectDocente

Quando o DUO pretende criar uma nova Unidade Curricular, este terá de selecionar o docente responsável da mesma, também terá que fazer este processo ao lançar novos DPUCs ou quando for necessário alterar o responsável de uma UC por qualquer motivo. Para facilitar a seleção de docentes, foi criado um componente onde se pode pesquisar e selecionar um docente através do seu número mecanográfico, nome ou email. Este componente é apresentado numa janela sobreposta para evitar navegações a outros endereços, mantendo uma experiência fluída no processo.

Selecionar Docente

Insira o Nome, Email ou Número Mecanográfico do docente

NMec	Nome	Email	
10311482	António Nuno Rosmaninho Rolo	rosmaninho@ua.pt	<button>Selecionar</button>
10314156	João Manuel de Oliveira e Silva Rodrigues	jmr@ua.pt	<button>Selecionar</button>
10312973	Maria Raquel Rocha Pinto	raquel@ua.pt	<button>Selecionar</button>
10310285	Maria Helena Abreu Silva	hsilva@ua.pt	<button>Selecionar</button>
10311839	Augusto Luis Barros Lopes	augusto.dr@ua.pt	<button>Selecionar</button>

< 1 >

Docente Selecionado

NMec	Nome	Email
10312280	Natália da Costa Martins	natallia@ua.pt

Guardar Alterações

Figura 27 Pesquisa de Docentes

Tal como na listagem de DPUCs existe uma paginação nos resultados para evitar listas demasiado longas. Este componente está desenvolvido de forma que, se for necessário selecionar vários docentes, terá de apenas “informar” o componente que pretende o mesmo (exemplo: um Regente escolher todos os docentes que lecionam uma Unidade Curricular).

Campo de Texto de um Formulário – Input

Para obter um campo de texto mais flexível (Campos maiores ou apenas com números) para os nossos formulários, criámos um componente com estilo idêntico à biblioteca do PACO2

Objetivos de aprendizagem* ^

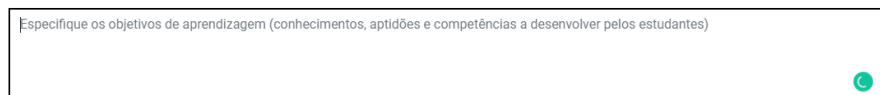


Figura 28 Campo de Texto do tipo "textarea"

Este tem o comportamento igual a um *input* de HTML normal.

Seleccionador de um Formulário – Selector

Dado que os componentes do PACO relativos a campos de seleção têm problemas na sua funcionalidade e com características em falta (exemplo: seleccionar várias opções) também foi desenvolvido um componente para seleccionar opções tendo em conta os estilos do componente correspondente na biblioteca do PACO.

Filtrar DPUCs por Unidade Orgânica:

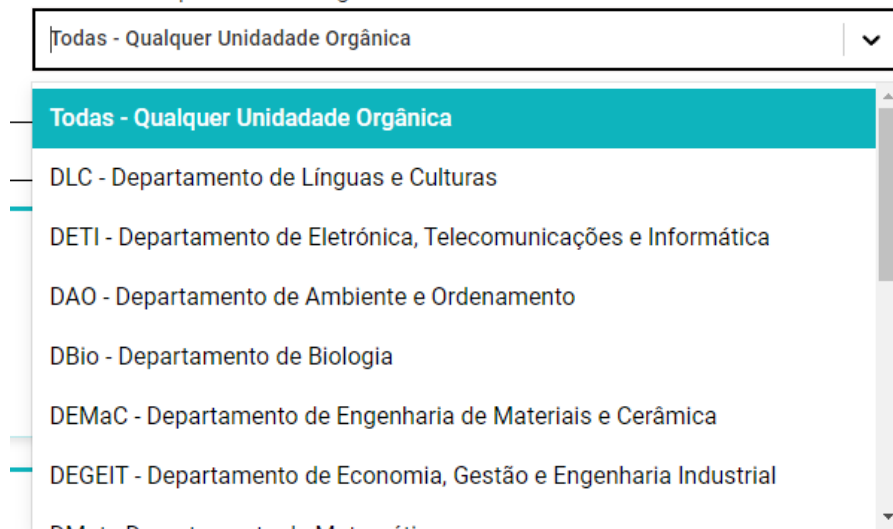


Figura 29 Seleccionador num formulário

Dashboards

Os *Dashboards* são as páginas iniciais para os diferentes tipos de utilizador, estes componentes são muito simples apenas possuindo uma breve descrição da página, no entanto se necessário será sempre fácil de adicionar conteúdo. Este componente faz uso da listagem de DPUCs.

Notícias | Agenda

[SGA] SGA | sair

universidade de aveiro paco

Gestão de DPUCs

Olá, SGA

Iniciar Processo De Edição

Filtrar DPUCs por Unidade Orgânica:
Todas - Qualquer Unidade Orgânica

Pesquisa de DPUC:
Pesquisa por nome, código ou docente regente

Filtrar DPUCs por estado:
Mostrar Todos (1340)

[44535] DLC - CULTURA PORTUGUESA CONTEMPORÂNEA	Aprovado (5)
Regente: António Nuno Rosmaninho Rolo	Edição de 2021/2022
Ultima alteração: 2021-09-01	
DPUC Aprovado.	Ver DPUC
[40379] DETI - FUNDAMENTOS DE PROGRAMAÇÃO	Aprovado (5)
Regente: João Manuel de Oliveira e Silva Rodrigues	Edição de 2021/2022
Ultima alteração: 2021-09-01	
DPUC Aprovado.	Ver DPUC
[42709] DAO - ÁLGEBRA LINEAR E GEOMETRIA ANALÍTICA	Aprovado (5)
Regente: Maria Raquel Rocha Pinto	Edição de 2021/2022
Ultima alteração: 2021-09-01	
DPUC Aprovado.	Ver DPUC
[40352] DBIO - BIOLOGIA CELULAR	Aprovado (5)

Figura 30 Dashboard para utilizadores dos SGA

Páginas

Criar Unidade Curricular – CreateUC

Nesta página é onde o Diretor de Unidade Orgânica inicia o processo de criação de uma Unidade Curricular e a primeira edição do DPUC correspondente. Esta página faz uso de componentes falados anteriormente (*Input*, *Selector* e *SelectDocente*) tais como outros componentes do PACO.

Figura 31 Página de criação de uma UC

Login – LoginPage

Para os utilizadores entrarem na nossa plataforma, devido à impossibilidade de integrar o IDP da Universidade foi criada uma página muito simples para autenticação.

Figura 32 Página de Login

Página Não Encontrada – NotFound

Também desenvolvemos uma página para endereços desconhecidos.

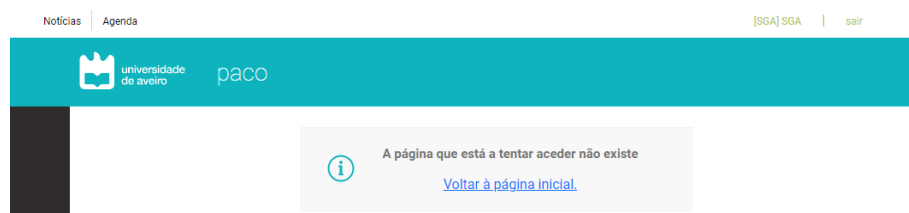


Figura 33 Página não encontrada

Editar DPUC – EditDPUC

Como uma das páginas principais do nosso projeto, após uma pesquisa sobre os campos que podem ser editados, desenvolvemos uma página de edição completa de um DPUC onde se podem alterar alguns campos, ficando os campos inalteráveis apenas visíveis. Esta página engloba diversos componentes visuais desenvolvidos e também da própria biblioteca do PACO 2.

The screenshot displays the 'Editar Dossier Pedagógico' form. At the top, there are buttons for 'Voltar', 'Submeter Para Aprovação', and 'Guardar'. The form is organized into several sections: 'Nome da Unidade Curricular' (Cultura Portuguesa Contemporânea), 'ECTS' (6), and 'Código' (44535); 'Unidade Orgânica' (Departamento de Línguas e Culturas) and 'Docente Responsável (Regente)' (10311482 - António Nuno Rosmaninho Rolo); 'Carga Letiva Semanal (em Horas)*' with input fields for TP (3), T (0), P (0), and OT (0); 'Duração*' (Semestral) and 'Semestre' (Primeiro); 'Modalidade de Lecção*' (dropdown) and 'Idioma(s) de leção*' (Português); and 'Página Pública da UC' (text input). A note at the bottom states 'Campos assinalados com * são obrigatórios'.

Figura 34 Edição de DPUC - Alguns campos pequenos, alteráveis

Objetivos de aprendizagem* ^

I – A descoberta de Portugal na transição para o século XX
 Apresentar o processo de desenvolvimento histórico da consciência nacional;
 Problematicar os discursos sobre a identidade nacional portuguesa produzidos nos séculos XIX e XX;
 Enunciar o contributo de alguns autores e movimentos intelectuais para o debate sobre a identidade nacional.
 Apreciar o crescente interesse pelo conhecimento directo e generalizado de Portugal, ocorrido na transição para o século XX-

II – Antropologia nacional
 Traçar o nascimento da etnografia em associação com a valorização da cultura popular e o desenvolvimento da consciência nacional;
 Verificar o processo de estetização da cultura tradicional inerente à valorização desse património;
 Observar o processo de estetização da cultura tradicional induzido pelo neogarretismo e promovido pelo Estado Novo;
 Integrar os ranchos folclóricos, as marchas populares e o concurso da «aldeia mais portuguesa de Portugal» (1938) no âmbito do processo de estetização da cultura tradicional;
 Identificar e criticar a interpretação caracterológica da identidade nacional;
 Expor e criticar a ideia de uma «personalidade-base» do Português, apresentada por Jorge Dias;
 Integrar o trabalho de Mendes Correia no contexto da antropologia anterior à II Guerra Mundial;
 Apresentar criticamente as conclusões de Mendes Correia quanto à existência de uma identidade biológica de Portugal.

III – Artes portuguesas
 Descrever o nascimento e o desenvolvimento da preocupação nacional na música, nas artes plásticas e na historiografia artística;
 Avaliar os fundamentos ideológicos, artísticos e etnográficos da «casa portuguesa» e das outras «casas nacionais»;
 Observar o processo de essencialização dos discursos nacionais;
 Identificar os principais tópicos de debate suscitados pela interpretação nacional da arte;
 Caracterizar artística e ideologicamente o neomanuelino e o neo-românico;

Pré-requisitos da UC* ^

Nenhum

Conteúdos programáticos* ^

I – A descoberta das culturas nacionais.
 II – Antropologia nacional: Romantismo e folclore; estetização da cultura tradicional; leitura caracterológica da identidade nacional; Jorge Dias; Mendes Correia.
 III – Artes portuguesas: a busca de uma arte portuguesa; manuelino e neomanuelino; «casa portuguesa»; classicismo monumental totalitário;

Figura 35 Edição de DPUC - Campos com grande área

Os campos com altas dimensões têm altura dinâmica, até um certo valor para não estender o tamanho de uma página, inicialmente serão baixos e dependendo do conteúdo aumentam. Os campos em que são esperados mais conteúdos terão uma altura maior naturalmente.

Ver DPUC – ViewDPUC

A página para ver um DPUC foi altamente baseada nas páginas de Unidades Curriculares do Portal da UA com os estilos do PACO 2 aplicados e sempre seguindo as normas do *Design System* da Universidade. Além das informações do DPUC serem apresentadas, também podem ser tomadas ações sobre o DPUC dependendo do estado e do tipo de utilizador.

[Notícias](#) | [Agenda](#) [SGA] SGA | [sair](#)

 universidade do aveiro **paco**

Cultura Portuguesa Contemporânea

[Voltar](#) [Vista Detalhada](#) Edição de 2022/2023 [Ações do DPUC](#)

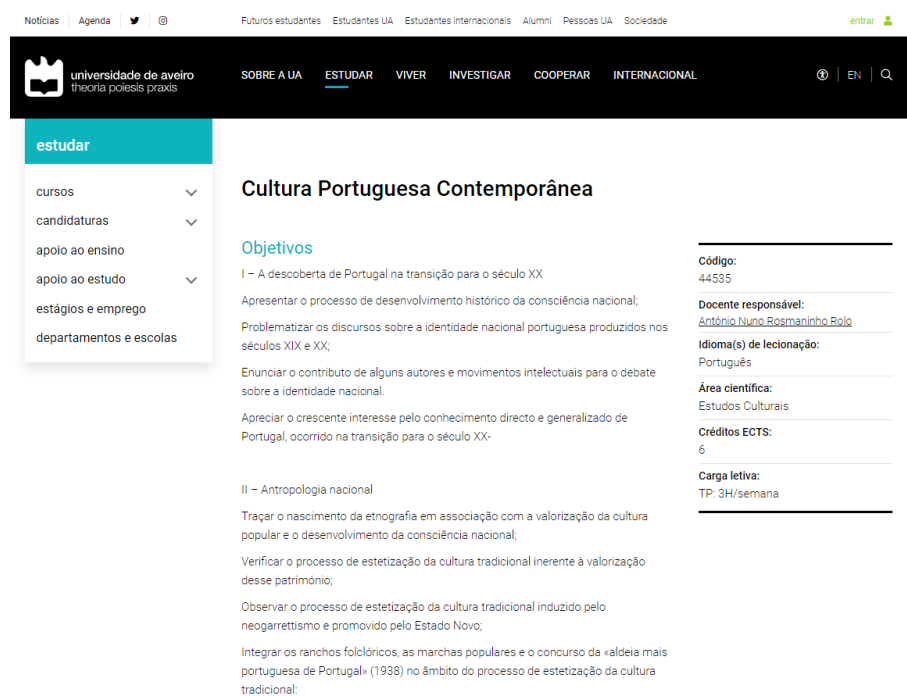
Última alteração: 11/06/2022

Objetivos

I – A descoberta de Portugal na transição para o século XX
Apresentar o processo de desenvolvimento histórico da consciência nacional;
Problematizar os discursos sobre a identidade nacional portuguesa produzidos nos séculos XIX e XX;
Enunciar o contributo de alguns autores e movimentos intelectuais para o debate sobre a identidade nacional.
Apreciar o crescente interesse pelo conhecimento directo e generalizado de Portugal, ocorrido na transição para o século XX-
II – Antropologia nacional
Traçar o nascimento da etnografia em associação com a valorização da cultura popular e o desenvolvimento da consciência nacional;
Verificar o processo de estetização da cultura tradicional inerente à valorização desse património;
Observar o processo de estetização da cultura tradicional induzido pelo neogarretismo e promovido pelo Estado Novo;
Integrar os ranchos folclóricos, as marchas populares e o concurso da «aldeia mais portuguesa de Portugal» (1938) no âmbito do processo de estetização da cultura tradicional;
Identificar e criticar a interpretação caracterológica da identidade nacional;

Código	ECTS
44535	6
Unidade Orgânica	
Departamento de Línguas e Culturas	
Área Científica	
Estudos Culturais	
Docente Responsável	
António Nuno Rosmaninho Rolo	
Idioma(s) de lecionação	
Português	
Modalidade	
Carga letiva semanal	
TP: 3H	
Duração	
Semestral	

Figura 36 Vista de DPUC



Notícias | Agenda | Futuros estudantes | Estudantes UA | Estudantes internacionais | Alumni | Pessoas UA | Sociedade

universidade de aveiro
theoria poiesis praxis

SOBRE A UA | **ESTUDAR** | VIVER | INVESTIGAR | COOPERAR | INTERNACIONAL

estudar

- cursos
- candidaturas
- apoio ao ensino
- apoio ao estudo
- estágios e emprego
- departamentos e escolas

Cultura Portuguesa Contemporânea

Objetivos

I – A descoberta de Portugal na transição para o século XX

Apresentar o processo de desenvolvimento histórico da consciência nacional;

Problematicar os discursos sobre a identidade nacional portuguesa produzidos nos séculos XIX e XX;

Enunciar o contributo de alguns autores e movimentos intelectuais para o debate sobre a identidade nacional.

Apreciar o crescente interesse pelo conhecimento directo e generalizado de Portugal, ocorrido na transição para o século XX-

II – Antropologia nacional

Traçar o nascimento da etnografia em associação com a valorização da cultura popular e o desenvolvimento da consciência nacional;

Verificar o processo de estetização da cultura tradicional inerente à valorização desse património;

Observar o processo de estetização da cultura tradicional induzido pelo neogarritismo e promovido pelo Estado Novo;

Integrar os ranchos folclóricos, as marchas populares e o concurso da «aldeia mais portuguesa de Portugal» (1938) no âmbito do processo de estetização da cultura tradicional.

Código:
44535

Docente responsável:
António Nuno Rosmaninho Rolo

Idioma(s) de leção:
Português

Área científica:
Estudos Culturais

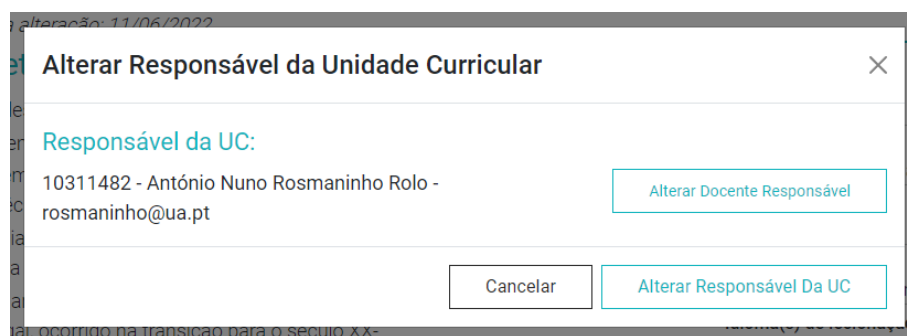
Créditos ECTS:
6

Carga letiva:
TP: 3H/semana

Figura 37 Vista de uma UC no Portal da UA

Ações

Como foi descrito ao longo do relatório, os diferentes atores podem fazer alterações ao estado de um DPUC, estas funcionalidades estão presentes como componentes de ação na nossa aplicação. Estes componentes servem tanto para confirmação da ação como para alterar algum dado necessário.



alteração: 11/06/2022

Alterar Responsável da Unidade Curricular

Responsável da UC:

10311482 - António Nuno Rosmaninho Rolo -
rosmaninho@ua.pt

Alterar Docente Responsável

Cancelar

Alterar Responsável Da UC

Figura 38 Ação: Alterar o Docente Responsável de uma UC

Componentes Auxiliares

Os próximos componentes comportam-se como funções de apoio ao correto funcionamento da aplicação.

Efetuar pedidos a uma API – useFetch

Este componente engloba o uso de criação de pedidos GET usando o cliente HTTP *Axios*. Sempre que utilizado é dado um *URL* e uma variável de controlo para repetir o pedido sempre que necessário. O componente retorna os dados, erros possíveis e informa do carregamento.

Obter Entidades – useGetAPIEntities

Nesta larga função obtém-se todas as entidades necessárias para o funcionamento da aplicação, estas entidades envolvem: Unidades Orgânicas, Áreas Científicas, Cursos, Estados de DPUC, etc. Estas entidades persistem em memória no contexto da aplicação desaparecendo assim que o utilizador sai da aplicação.

Obter DPUC – useGetDPUC

Esta função trata de organizar o conteúdo de um DPUC transmitido pela API do *Fetcher* transformando os ids de algumas entidades nos objetos respondentes e formatando alguns dados para ficarem mais fáceis de ler, por exemplo o campo das línguas do DPUC vem no formato “Português\$Inglês\$”, o que não é facilmente lido para os utilizadores.

Traduzir dados para DPUC – getFormattedDPUC

Ao contrário da última função, esta converte um objeto de um DPUC para um formato aceitável pela API do *Fetcher* convertendo objetos de entidades nos ids das mesmas e compacta outros campos para existir menor uso de largura de banda nas redes.

Gestão de JWT – useUserData

Esta é uma função responsável pela gestão dos *tokens* de autorização para acesso à API do *Fetcher*. Assim que a aplicação inicia procura nas cookies pelo *token* e o utilizador correspondente também ficando responsável pela alteração das mesmas caso a sessão expire, seja feito um *logout* ou alteração de utilizador.

Problemas e Desafios

Nesta secção serão descritos os maiores problemas encontrados no desenvolvimento da aplicação *web* e como conseguimos resolver ou contornar alguns desses problemas.

Biblioteca de Componentes do PACO 2

Instalação

Devido à forma que a biblioteca em *React.JS* do PACO 2 foi desenvolvida foi necessário estudar o funcionamento do *Node.js* e da forma que o *React* gere os seus pacotes e dependências. A integração desta biblioteca gerou problemas pois vários módulos internos foram desenvolvidos em versões de *React* e *WebPack* diferentes e, para combater estes conflitos, a versão do *React.JS* e alguns módulos (*WebPack* e *React Router*, por exemplo) foram rebaixadas sendo que algumas destas versões já tinham sido descontinuadas criando uma má prática no desenvolvimento da aplicação. Após a alteração da versão dos módulos, foi necessário reestruturar a raiz da aplicação para o correto funcionamento e também remover diversos *bugs* que surgiram. Além dos problemas com as versões, também surgiu a necessidade de chaves de API com uma das dependências do módulo do PACO: *Font-Awesome* o que, para adicionar ao projeto também foi complicado para a configuração e exportação da mesma.

Componentes

Ao longo do desenvolvimento do projeto foram detetadas algumas inconsistências e problemas com alguns dos componentes do PACO 2 sendo estes em específico os campos de texto e de seleção. As inconsistências encontradas são relativas a componentes parecidos aos nativos de *React.JS* ou do *Bootstrap* que, no caso dos do PACO têm opções de customização muito limitadas tal como algumas funcionalidades não estarem bem documentadas (PACO 2 Storybook, s.d.).

Um exemplo mais concreto é o uso do componente `<Text>` onde os *props* (variáveis do componente) *color* não aceitam qualquer valor de cor em RGB, hexadecimal ou nome de cor em CSS, mas sim cores do seu próprio tema como por exemplo “*primary*”, “*success*”, etc. Estas cores não estão indicadas na documentação dos componentes, pelo que ficam apenas subentendidas. Também acontece que o *prop* “*fontWeight*” não aceita qualquer valor inteiro, mas sim valores com intervalos muito espaçados (aceita os valores 200, 400 e 700, mas não valores intermediários).

Também foram encontrados problemas com os componentes de introdução de texto e de seleção de itens, estes eram muito flexíveis e não funcionavam corretamente. Os desenvolvedores foram contactados e alertados relativamente a este problema, mas também foram incapazes de os resolver tal como as inflexibilidades.

Criar nova UC

Nome da Unidade Curricular

Unidade Orgânica

Docente Responsável (Regente)

ECTS

Figura 39 Problemas num campo de seleção

No entanto, como desenvolvedores da aplicação *web* não podíamos ficar dependentes da resolução dos problemas de componentes externos e para tal, foram criados 2 componentes: *Selector* e *Input* que já foram descritos anteriormente. Estes novos componentes criados têm mais flexibilidade e podem ser facilmente adicionados a qualquer outro projeto.

Barra de Navegação

Para tornar a nossa plataforma mais familiar para os utilizadores, solicitámos os componentes relativos às barras de navegação do PACO 2 ou as suas especificações visuais, porém, este pedido foi nos recusado. Como ainda assim pretendíamos adicionar um certo nível de realismo ao projeto as barras de navegação do PACO foram estudadas e criados componentes visualmente semelhantes.

Notícias | Agenda

[SGA] SGA | sair

universidade do aveiro

paco

Gestão de DPUCs

Olá, SGA

Iniciar Processo De Edição

Filtrar DPUCs por Unidade Orgânica:

Todas - Qualquer Unidade Orgânica

Pesquisa de DPUC:

Pesquisa por nome, código ou docente regente

Filtrar DPUCs por estado:

Mostrar Todos (2677)

Figura 40 Barras de Navegação no nosso projeto

Testes e Comentários

Devido a limitações de tempo não foram concretizados testes de usabilidade. No entanto, ao longo do projeto existiram algumas demonstrações a docentes e funcionários da Universidade e, consoante sugestões e críticas, foram tomadas medidas no desenho da aplicação *web* existindo uma evolução no percurso do projeto.

Estas foram as fases, a nível de *design*, mais relevantes:

- Protótipo não funcional com *Proto.io*
 - Criação de algumas páginas seguindo o formato das páginas do Portal da UA.
- Protótipo funcional com *React.JS*
 - Implementação das páginas do protótipo anterior seguindo o Design System da Universidade usando uma biblioteca de componentes do Portal da UA.
- Aplicação *Web* com Biblioteca de Componentes em *React.JS* do PACO
 - Evolução do protótipo para passar a usar o *Fetcher* desenvolvido, implementando componentes usados no website do PACO 2 removendo os componentes da biblioteca do Portal da UA.

8.4. Autenticação

Backend

Devido à falta de conhecimento no desenvolvimento de um sistema de autenticação, fizemos a nossa pesquisa de qual seria a melhor forma de implementar e acabamos por nos basear num esqueleto desenvolvido pelo blog JavaInUse (JavaInUse, 2019).

Sendo a prova de conceito desenvolvida *role based*, utilizamos um sistema de autenticação baseado em JSON Web Token (JWT), um *open standard* usado para partilhar informação de uma forma segura entre duas entidades. Deste modo, com este tipo de login conseguimos garantir a autenticação não só no momento de iniciar sessão, bem como em cada endpoint, pois o token de autenticação é enviado em cada pedido no *Autorization Header*, sendo o tipo de token um *Bearer Token*.

Deste modo, para realizar o iniciar de sessão o utilizador faz uso do seguinte endpoint:

login-controller : Login Controller Show/Hide List Operations Expand Operations

POST /authenticate createAuthenticationToken

Response Class (Status 200)
OK

Model Example Value

Inline Model {}

Response Content Type */*

Parameters

Parameter	Value	Description	Parameter Type	Data Type
authenticationRequest	<pre>{ "username": "sga@ua.pt", "password": "123" }</pre>	authenticationRequest	body	Model Example Value <pre>{ "password": "string", "username": "string" }</pre>

Parameter content type: application/json

Figura 41 Documentação Swagger

Quando é dado um username e password corretos é gerado o JSON Web Token, sendo este gerado pelo seguinte fluxo (o JwtAuthenticationController o nosso LoginController):

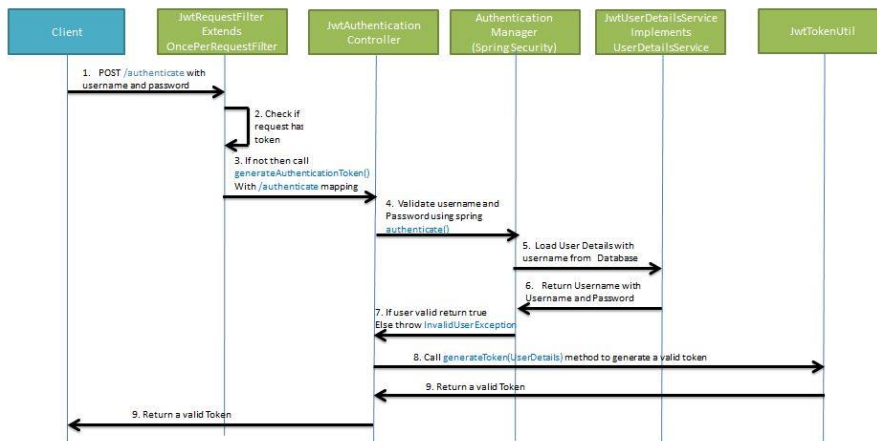


Figure 1 Generating JWT Diagram (JavaInUse)

O AuthenticationManager é uma interface do Spring Security que faz uso da função authenticate, retornando true se o username e password estiverem na base de dados, caso contrário levanta a exceção DisabledException ou BadCredentialsException.

Para efetuar a validação do token nos endpoints, existe o seguinte fluxo desde o cliente até ao controlador:

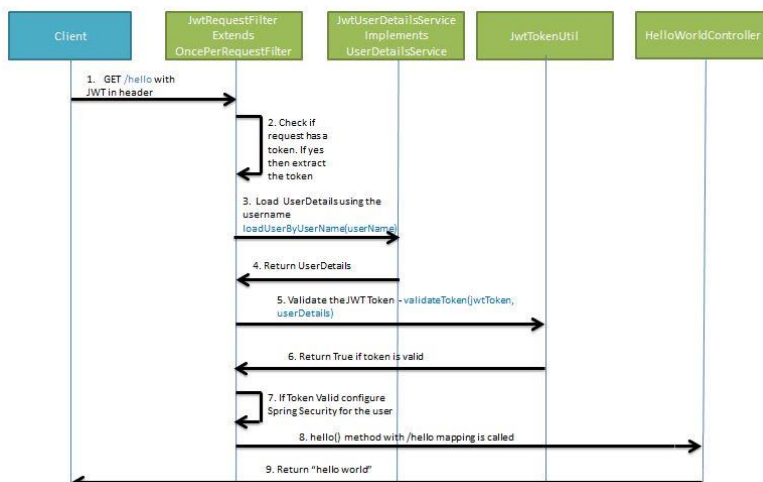


Figure 2 Validating JWT (JavaInUse)

Explicação dos ficheiros***LoginController (JwtAuthenticationController)***

Recebe o username e password no body e faz uso da função authenticate do AuthenticationManager para validar a tentativa de login. Sendo válida gera o token, caso contrário lança uma exceção.

application.properties

É definido um uma chave secreta que é combinada com o header e o payload para criar uma hash única, hash essa que é só possível de ser verificada se for definido este segredo.

JwtTokenUtil

Esta classe é responsável por realizar operações JWT como a criação e a validação.

JWTUserDetailsService

Implementa o Spring Security UserDetailsService interface, dando override ao método loadUserByUsername que vais buscar o utilizador à base de dados dado um email. Esta função é crucial, pois é utilizada pelo AuthenticationManager.

JwtRequest

Esta classe para armazenar o username e password do utilizador.

JwtResponse

Esta classe de resposta para utilizador, contendo o JWT necessário para o utilizador navegar na aplicação.

JwtRequestFilter

Esta classe deriva da classe Spring Web Filter OncePerRequestFilter. Para todos os pedidos feitos aos endpoints este filtro é executado para validar o token.

JwtAuthenticationEntryPoint

Esta classe deriva do Spring AuthenticationEntryPoint e faz override do método commence com o objetivo de rejeitar todos os pedidos não autenticados.

WebSecurityConfig

Esta classe deriva da WebSecurityConfigurerAdapter e é uma classe de que permite a personalização do WebSecurity e HttpSecurity.

Frontend

Na aplicação *web* a autenticação é feita de forma simplificada para o utilizador final. Primeiramente o utilizador acede à página de login onde faz a autenticação, de seguida é redirecionado para a *Dashboard* correspondente. Internamente, ao fazer a autenticação, o *browser* recebe um token e informações sobre o utilizador e estes dados ficam guardados em cookies, desta forma se o utilizador atualizar a página ou fechar e abrir o *website* num curto espaço de tempo não perderá a sessão pois a aplicação verificará se existem dados nas *cookies* e se sim, não será necessário fazer um novo login. Ao fazer o logout, as *cookies* relativas à sessão são apagadas e o utilizador ficará com a página de login em disposição até fazer um novo login.

8.5. Pesquisa

As implementações iniciais dos motores de busca foram feitas usando 3 componentes principais: Um nó do *Elasticsearch*, um servidor rest-api *Django*, e uma plataforma do tipo Kibana.

O servidor Django faz a conexão entre a base de dados e o nó do *Elasticsearch*, obtendo os dados considerados relevantes das unidades curriculares, convertendo em uma estrutura do tipo JSON par chave-valor, em que a chave é o id da unidade curricular, e o valor é um dicionário que contém os restantes dados. Esses dados são inseridos em um índice específico que será usado para fazer a pesquisa por texto. Além disso, o servidor é responsável por regularmente atualizar os dados do índice a partir das mudanças na base de dados. Para isso, o mesmo possui um mecanismo Time-to-live e um endpoint que notifica o mesmo de que houve mudanças na base de dados e o mesmo deve atualizar o índice.

O servidor possui dois endpoints:

- /search

O endpoint que é usado para requisitar a pesquisa, retorna uma lista de códigos das unidades curriculares consideradas relevantes. Possui como parametro opcional keywords, que deve ser uma lista de palavras que irão definir os valores retornados pelo critério de relevância. Caso dê tudo certo, retorna uma resposta de status 200, senão, retorna uma resposta de status 500, que representa erro interno no servidor, possivelmente relacionado a conexão com a base de dados e/ou nó do Elasticsearch, ou a manipulação dos dados

- /update

Endpoint que não necessita de argumentos e que notifica o servidor de que houve atualização na base de dados. Se a atualização corre bem, retorna uma resposta de status 200, a notificar que deu tudo certo. Caso contrário, recebe uma resposta do tipo 500, que representa erro interno do servidor, possivelmente pelos mesmos motivos citados anteriormente.

Os seguintes campos são requisitados pela base de dados e armazenados no Elasticsearch. Eles foram escolhidos por serem considerados importantes em termos de

relevância na pesquisa pelos utilizadores, exceptuando o Id, que é necessário para representar a entidade de dados de uma Unidade Curricular:

- Id
- Nome
- Objetivos
- Conteúdos

A componente do Elasticsearch é que é responsável pelos mecanismos de pesquisa baseados em relevância através de processamento de texto. Além de retornar resultados relevantes para uma requisição, ordena os mesmos baseado em um score que representa o grau de relevância do resultado para a busca. Geralmente, o Elasticsearch é distribuído e replicado em grupos de cluster, porém considerando o tamanho pequeno de dados a processar, consideramos que apenas um nó, sem replicação interna, é o suficiente para garantir um funcionamento eficiente da nossa aplicação.

Além de ambas as componentes citadas anteriormente, também temos uma plataforma do tipo Kibana, que serve para administrar o funcionamento do nosso Elasticsearch, além de conseguir obter métricas relacionadas ao uso e gestão dos dados, como a média de tempo em que os documentos de um índice específico é atualizado, número médio de documentos deletados em determinado intervalo de tempo e até mesmo número de requisições por pesquisa, que no nosso contexto, provém do servidor a pedido dos utilizadores.

O mecanismo de busca por relevância é baseado em dois requisitos principais: resultados relevantes relativamente precisos e resultados ordenados decrescentemente por um valor score de relevância. O mecanismo que implementamos é básico e é baseado em 3 campos: Nome, Conteúdos e Objetivos. O parametro da busca passado para o Elasticsearch é ilustrado abaixo em formato JSON.

```
{ "bool":  
  { "should": [  
    { "match": { "nome": keywords } },  
    { "match": { "conteudos": keywords } },  
    { "match": { "objetivos": keywords } }  
  ] }  
}
```

O que o documento representa são múltiplas *queries*, que aumentam o escopo de resultados retornados, que são garantidos pelo campo “**bool**”. Já o campo “**should**” diz ao

Elasticsearch para retornar qualquer resultado que coincida com pelo menos uma das *queries*. A ordem das mesmas é importante para garantir o valor de score do resultado. Por exemplo, um resultado que coincida com o campo nome, terá mais valor que um resultado que coincida com conteúdos.

Consideramos que o mecanismo de busca possa ser mais bem desenvolvido posteriormente, o que inclui o processamento sintático da keywords para, por exemplo, perceber se a mesma representa uma frase ou uma lista de palavras independentes. Além disso, é possível alterar a keywords para obter palavras semelhantes que tornem a busca mais alargada, como obter a palavra português ou portuguesa a partir da palavra Portugal que foi passada pelo utilizador para a busca. Todas estas funcionalidades envolvem uso de processamento de linguagens naturais, tópico académico a qual o grupo não tem conhecimentos suficientes para implementar. Seu uso pode ser feito se utilizando da biblioteca do Spacy do Python. Além dessas funcionalidades, queríamos ter podido implementar múltiplos mecanismos de ordenação, além do da relevância, como ordem alfabética ou data de criação.

Infelizmente, devido a questões de configuração, tal componente de pesquisa não pode ser posta em produção, porém já existe uma base para no futuro esta funcionalidade ser implementada e integrada com o projeto.

9. Conclusões e Trabalho Futuro

9.1. Resultados Práticos

De uma forma geral, apesar das dificuldades no estabelecimento dos requisitos e processos, foi atingido com sucesso o bom funcionamento do fluxo tanto no processo de Criação de um primeiro DPUC, bem como no processo de Edição de um DPUC de uma UC já aprovada (em estado ativo).

Abstraimo-nos de entidades administrativas no processo de aprovação de uma DPUC acoplamos essas mesmas nos SGA, sendo em termos práticos o nosso administrador. Deste modo, são quem pode voltar atrás nos estados para corrigir possíveis erros, aprovar DPUCs nos processos, desativar DPUCs e dar abertura ao processo de Edição.

De forma a compreender os resultados práticos em ambos os processos, relembramos os diferentes estados de cada um.

É de notar, que o estado de edição E3 foi desenvolvido para o Diretor de Curso em que, quando campos sensíveis são editados, existir a necessidade de aprovar para passar para o estado E4. Seguindo a orientação do professor José Vieira deixamos este estado inativo, passando assim do E2 diretamente para o E4. De qualquer das formas, existe já o endpoint desenvolvido no controlador, bem como a sua lógica implementada no serviço.

Estados Criação		Estados Edição
C1 – Em criação		E1 – Definição de DUO
C2 – Em edição		E2 – Em edição
C3 - Fechado		E3 – Em aprovação (INATIVO)
C4 – Em aprovação		E4 – Aprovado
C5 - Aprovado		
C6 - Desativado		

Criação

O DUO pode criar uma nova UC (estado C1), submeter a primeira versão do DPUC para ser posteriormente completada pelo DR (estado C2).

O DR pode editar DPUCs em aberto (estado C2) e submeter quando finaliza a sua edição (estado C3 - fechado).

Os SGA podem passar o DPUC em estado C3 para o estado C4 (em aprovação) atribuindo-lhe um código. Também podem passar para o estado C5 (aprovado), ou passar para o estado C2 se detetarem algum erro. Em último caso, podem desativar passando para o estado C6.

Edição

Quando os SGA iniciam o processo de edição é criada uma cópia para todos os DPUCs que estão no estado C5 ou E4, metendo-os no estado E1 para iniciar o processo “oficialmente”.

O DUO pode confirmar o DR anterior ou seleccionar um novo no estado E1, quando este define-o passa para o estado E2.

O DR atribuído ao DPUC pode visualizar o DPUC do ano anterior e fazer as alterações que achar necessárias. Quando acabar a edição pode submeter passando automaticamente para o estado E4.

É de notar que os SGA podem reabrir o DPUC passando-o para o estado E2 e o DUO a qualquer momento pode alterar o DR (desenvolvido para casos especiais, ex.: o DR reforma-se ou fica incapacitado de dirigir a UC).

Outras funcionalidades

Ver a lista de DPUCs de UCs ativas, visualização detalhada de uma DPUC, acesso a diferentes versões do DPUC e filtrar DPUCs por estado, nome ou código.

O DUO pode ver todos os DPUCs das UCs da sua Unidade Orgânica.

O DR pode ver todos os DPUCs das UCs das quais é responsável.

10. Conclusões e execução do projeto

Para concluir, tendo em conta o trabalho realizado, as dificuldades encontradas e os resultados obtidos, concluímos este projeto com um bom grau de satisfação e acreditamos que poderá ser uma forte base de referência para continuar num futuro desenvolvimento na área, com o objetivo de a nossa plataforma ser a futura plataforma de manutenção de DPUCS da UA. No entanto, como qualquer informático, por melhor que esteja um projeto desenvolvido ficamos sempre com a ideia que poderíamos ter desenvolvido a plataforma mais profundamente, mesmo que o tempo e todos os obstáculos não o tenham permitido.

Execução do projeto

Localmente

Pré-requisitos: *Node.js* instalado com Base de dados e Backend funcionais.

- No diretório base “DPUC/” executar no terminal:
 - *\$docker-compose up --build*
- No diretório “DPUC/teacher-app/” para instalar as dependências necessárias a aplicação *web*:
 - *\$npm install*
 - *\$npm start*
- Se, devido a problemas com as dependências externas, os comandos *npm* não estiverem funcionais, antes de voltar a tentar experimente [1] e se problemas persistirem experimente [2]
 1. *\$npm cache clear --force*
 2. *\$npm install --force*

Após começar o servidor *React* este ficará disponível em <http://localhost:3000/>.

Para uso da aplicação, existem diversos utilizadores disponíveis, para facilidade todas as palavras-passe são “123”, Disponibilizaremos aqui um utilizador para cada ator (remover os espaços no email)

- SGA
 - sga @ ua . pt
 - 123
- DUO
 - jtorrao @ ua . pt
 - 123
- DR
 - rosmatinho @ ua . pt
 - 123

Azure Cloud

Ou podem também aceder ao link <http://dpuc-ua.uksouth.cloudapp.azure.com:81/> , pois o projeto foi colocado na azure.

Para efeitos de testes deixamos aqui uma conta para cada ator do sistema:

SGA – email: sga@ua.pt, password: 123

DUO - email: jtorrao@ua.pt, password: 123

DR - email: rosmatinho@ua.pt, password: 123

11. Trabalho Futuro

Em relação às próprias funcionalidades da aplicação, a exportação para PDF e outros formatos seria bastante útil para os professores, e exportação direta para o formato da A3ES. Além disso, ter um sistema de *checks* de visibilidade em alguns campos para o docente escolher o que quer apresentar para os alunos do DPUC, sendo que existe informação neles desnecessária e interna para os alunos. Seria interessante, tendo sido apontado por um docente nos testes ao sistema, a opção de ir buscar um dpuc anterior ao do ano passado no processo de edição, dar essa escolha ao docente pode ser bastante útil.

Em relação ao Motor de Pesquisa que desenvolvemos, embora não tenhamos conseguido configurar para funcionar por atrasos de implementação em outras partes, o mesmo ainda

pode ser no futuro configurado, e se possível melhorado. Os principais aspetos a melhorar são criar opções de filtragem, como por exemplo, Unidades Curriculares de determinado DR ou Unidade Orgânica, ordenação, como ordem alfabética e/ou por data de criação, além da ordenação por relevância, e especialmente, no desenvolvimento do mecanismo de relevância, usando-se de processamento de linguagens naturais, como por exemplo, vê se as palavras-chaves são sintaticamente uma frase ou palavras independentes. Além disso, é possível criar outras palavras-chaves a partir das introduzidas, recurso implementado com o uso de *word-clouds*, como por exemplo, a partir da palavra “Portugal” inserida pelo utilizador, gerar as palavras “português” e “portuguesa”, e aumentar o *recall* de resultados relevantes retornados.

12. Bibliografia

JavaInUse. (2019, May 8). *Spring Boot Security + JWT Hello World Example*. Retrieved from javainuse: <https://www.javainuse.com/spring/boot-jwt>

UA Design System. (s.d.). Obtido de <https://universidadeaveiro.github.io/portal-shared-library/>

React.JS. (s.d.). Obtido de <https://reactjs.org/>

React Bootstrap. (s.d.). Obtido de <https://react-bootstrap.github.io/getting-started/introduction/>

PACO 2 Storybook. (s.d.). Obtido de <http://paco20-storybook.s3-website-eu-west-1.amazonaws.com/?path=/story/appendix-paco-2-0-storybook--page>

PACO UI NPM. (s.d.). Obtido de https://www.npmjs.com/package/@paco_ua/pacoui

Axios. (s.d.). Obtido de <https://www.npmjs.com/package/axios>

React Router. (s.d.). Obtido de <https://v5.reactrouter.com/web/guides/quick-start>

13. Anexos

13.1. Anexo I

10/06/2022, 23:31 Swagger UI

Api Documentation

Api Documentation

[Apache 2.0](#)

edition-controller : Edition Controller

PUT

/edition/definicaoRegente

definicaoRegente

Response Class (Status 200)

int32

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="(required)"/>	id	query	integer
regenteid	<input type="text" value="(required)"/>	regenteid	query	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

[Try it out!](#)

PUT

/edition/emAprovacao

emAprovacao

10/06/2022, 23:31 Swagger UI

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

PUT /edition/emEdicao emEdicao

Response Class (Status 200)
string

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
json	(required) <div></div>	json	body	string
Parameter content type: application/json ▼				
id	(required)	id	query	integer
finished	true ▼	finished	query	boolean

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

GET /edition/getAllDpucs getAllDpucs

Response Class (Status 200)
OK

Model Example Value

localhost:82/swagger-ui.html#/edition45controller 2/15

10/06/2022, 23:31

Swagger UI

```
[
  {
    "aprendizagem": "string",
    "bibliografia": "string",
    "carga_horaria": "string",
    "coerencia_conteudos": "string",
    "coerencia_metodologias": "string",
    "conteudos": "string",
    "criacao_edicao": true,
    "cursos": "string",
  }
]
```

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	id	query	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

PUT /edition/iniciarEdicao

iniciarEdicao

Response Class (Status 200)

string

Response Content Type */* ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

PUT /edition/setRegente

setRegente

Response Class (Status 200)

localhost:82/swagger-ui.html#/edition45controller

3/15

10/06/2022, 23:31 Swagger UI

string

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	id	query	integer
regenteid	(required)	regenteid	query	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

login-controller : Login Controller

POST /authenticate createAuthenticationToken

Response Class (Status 200)
OK

Model | Example Value

```
{}
```

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------

10/06/2022, 23:31 Swagger UI

string

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="(required)"/>	id	query	integer
regenteid	<input type="text" value="(required)"/>	regenteid	query	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

[Try it out!](#)

login-controller : Login Controller

POST /authenticate createAuthenticationToken

Response Class (Status 200)
OK

Model | Example Value

```
{}
```

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------

10/06/2022, 23:31 Swagger UI

Parameter	Value	Description	Parameter Type	Data Type
authenticationRequest	(required)	authenticationRequest	body	Model Example Value

Parameter content type: application/json ▼

```
{  "password": "string",  "username": "string"}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

Response Class (Status 200)

OK

Model | Example Value

```
{}
```

Response Content Type */* ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		

localhost:82/swagger-ui.html#/edition45controller 5/15

10/06/2022, 23:31

Swagger UI

HTTP Status Code	Reason	Response Model	Headers
404	Not Found		

Try it out!

main-controller : Main Controller

GET

/creation/acs

getAreasCientificas

Response Class (Status 200)

OK

Model Example Value

```
[
  {
    "designacao": "string",
    "id": 0,
    "sigla": "string"
  }
]
```

Response Content Type */* ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

PUT

/creation/aprovarDpuc

aprovarDpuc

Response Class (Status 200)

string

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------

localhost:82/swagger-ui.html#/edition45controller

6/15

10/06/2022, 23:31

Swagger UI

Parameter	Value	Description	Parameter Type	Data Type
aprovado	(required) <div></div>	aprovado	body	string
Parameter content type: application/json ▼				
id	(required)	id	query	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

POST /creation/criarUC

criarUC

Response Class (Status 200)

string

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
uc	(required) <div></div>	uc	body	string
Parameter content type: application/json ▼				
regenteid	(required)	regenteid	query	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		

localhost:82/swagger-ui.html#/edition45controller

7/15

10/06/2022, 23:31 Swagger UI

HTTP Status Code	Reason	Response Model	Headers
403	Forbidden		
404	Not Found		

Try it out!

GET /creation/cursos getCursos

Response Class (Status 200)
OK

Model Example Value

```
[
  {
    "id": 0,
    "nome": "string",
    "unidade_organicaid": 0,
    "utilizadores_id": 0
  }
]
```

Response Content Type */* ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

GET /creation/cursos/uo getCursos

Response Class (Status 200)
OK

Model Example Value

```
[
  {
    "id": 0,
    "nome": "string",
    "unidade_organicaid": 0,
    "utilizadores_id": 0
  }
]
```

localhost:82/swagger-ui.html#/edition45controller 8/15

10/06/2022, 23:31 Swagger UI

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
uo	(required)	uo	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

PUT /creation/desativarDpuc desativarDpuc

Response Class (Status 200)
string

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	id	query	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

GET /creation/docentes getDocentes

Response Class (Status 200)
OK

Model Example Value

localhost:82/swagger-ui.html#/edition45controller 9/15

10/06/2022, 23:31 Swagger UI

```
[
  {
    "email": "string",
    "id": 0,
    "nmec": 0,
    "nome": "string",
    "password": "string",
    "tipo_utilizadorid": 0
  }
]
```

Response Content Type */* ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

GET /creation/dpucs getDpucs

Response Class (Status 200)
OK

Model Example Value

```
[
  {
    "aprendizagem": "string",
    "bibliografia": "string",
    "carga_horaria": "string",
    "coerencia_conteudos": "string",
    "coerencia_metodologias": "string",
    "conteudos": "string",
    "criacao_edicao": true,
    "cursos": "string",
  }
]
```

Response Content Type */* ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

localhost:82/swagger-ui.html#/edition45controller 10/15

10/06/2022, 23:31

Swagger UI

GET /creation/dpucs/{id} getDpuc

Response Class (Status 200)
OK

Model | Example Value

```
{
  "aprendizagem": "string",
  "bibliografia": "string",
  "carga_horaria": "string",
  "coerencia_conteudos": "string",
  "coerencia_metodologias": "string",
  "conteudos": "string",
  "criacao_edicao": true,
  "cursos": "string",
  "data_alteracao": "string",
}
```

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	id	path	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

PUT /creation/editarDpuc editarDpuc

Response Class (Status 200)
string

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------

localhost:82/swagger-ui.html#/edition45controller

11/15

10/06/2022, 23:31 Swagger UI

Parameter	Value	Description	Parameter Type	Data Type
dpuc	(required) <div></div>	dpuc	body	string
Parameter content type: application/json ▼				
id	(required)	id	query	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

PUT /creation/emAprovacao aprovarDpuc

Response Class (Status 200)
string

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	id	query	integer
codigo	(required)	codigo	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

10/06/2022, 23:31 Swagger UI

GET /creation/estados getEstados

Response Class (Status 200)
OK

Model **Example Value**

```
[
  {
    "descricao": "string",
    "id": 0,
    "nome": "string"
  }
]
```

Response Content Type */* ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

[Try it out!](#)

PUT /creation/fecharDpuc fecharDpuc

Response Class (Status 200)
string

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	id	query	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

localhost:82/swagger-ui.html#/edition45controller 13/15

10/06/2022, 23:31 Swagger UI

Try it out!

GET /creation/idiomas getIdiomas

Response Class (Status 200)
OK

Model Example Value

```
[  
  "string"  
]
```

Response Content Type */* ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

GET /creation/periodos-letivos getPeriodos

Response Class (Status 200)
OK

Model Example Value

```
[  
  {  
    "id": 0,  
    "periodo": "string"  
  }  
]
```

Response Content Type */* ▼

localhost:82/swagger-ui.html#/edition45controller 14/15

