



UNIVERSIDADE
LUSÓFONA

Plataforma DEISI – Gestão de TFCs (Análise e Testes)

Trabalho Final de curso

Diogo Cesar - 21703920

Rafael Horta - 21705375

Docente: Professor Pedro Alves

Trabalho Final de Curso | LEI | 26/04/2020

www.ulusofona.pt

Direitos de cópia

Plataforma DEISI – Gestão de TFCs (Análise e Testes), Copyright de Diogo Cesar e Rafael Horta, ULHT

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Resumo	5
Abstract	6
1. Identificação do problema	7
a. Inquérito de satisfação a utilizadores	7
b. Resultados	7
c. Conclusões	10
2. Viabilidade e pertinência	11
a. Factores econométricos	11
3. Levantamento e análise de requisitos	12
a. Requisitos Funcionais	12
b. Requisitos Não Funcionais	17
4. Solução desenvolvida	25
a. Enquadramento prático	25
b. Tecnologias utilizadas	27
c. Ambiente de desenvolvimento	27
i. Análise dos problemas documentados no repositório da plataforma	27
ii. Escrita de testes automáticos	30
d. Ambiente de produção e Test-Driven Development	31
5. Benchmarking	33
a. Análise comparativa	33
6. Plano de testes e validação	34
7. Resultados	46
a. Inquérito aos alunos	46
i. Resultados	47
b. Inquérito aos docentes	47
i. Resultados	47
8. Conclusões e trabalhos futuro	50
Bibliografia	51
Anexos	52
Glossário	54

Índice de figuras

Figura 1 - Casos de uso do Coordenador.	20
Figura 2 - Casos de uso do Docente Avaliador.	20
Figura 3 - Casos de uso do Aluno	21
Figura 4 - Listagem de problemas no repositório da plataforma.	28
Figura 5 - Comentário do professor Pedro Alves que deu a conhecer o problema.	29
Figura 6 - Proposta de resolução indicada pelo grupo após a correção.	30
Figura 7 - Falhas detetadas e ainda não corrigidas.	32

Índice de tabelas

Tabela 1 - Histórico de versões: UC 14	21
Tabela 2 - UC 14	24
Tabela 3 - Listagem de problemas.	53

Índice de gráficos

Gráfico 1 - Classifique o nível de intuição ao navegar pela plataforma.	7
Gráfico 2 - Classifique o nível de organização e estruturação da plataforma.	7
Gráfico 3 - Classifique o nível de consistência da interface.	8
Gráfico 4 - Classifique o nível de apelação da interface.	8
Gráfico 5 - Com quantos problemas se deparou ao utilizar a plataforma?	9
Gráfico 6 - Na sua opinião, a plataforma oferece todas as funcionalidades necessárias para o seu funcionamento ser adequado à UC?	9
Gráfico 7 - Deparou-se com problemas de performance?	10
Gráfico 8 - Conseguiu realizar o download dos trabalhos que lhe foram atribuídos?	48
Gráfico 9 - Conseguiu listar e ver as avaliações aos trabalhos que orienta, à medida que estas foram submetidas?	49

Resumo

A unidade curricular (UC) Trabalho Final de Curso (TFC), trata-se de “*um elemento fundamental do currículo das Licenciaturas em Informática de Gestão (LIG) e Engenharia Informática (LEI) da Universidade Lusófona de Humanidades e Tecnologias (ULHT) funcionando como um momento nuclear na formação dos alunos, nomeadamente em ordem à promoção da sua integração profissional e iniciação ao trabalho científico*”. [1]

A gestão desta UC é efetuada por meio de uma plataforma de gestão de *workflow* desenvolvida também para TFC, intitulada de “Plataforma DEISI - Gestão de TFCs” de Francisco Santos, inicialmente desenvolvida no ano lectivo 2017/2018.

A plataforma resume-se numa aplicação web em *Spring MVC (Model-View-Controller)* que permite “*que todos os intervenientes consigam executar as tarefas, de forma fluída e respeitando o normal workflow definido no respetivo regulamento*”. [2]

O presente trabalho teve como objetivo dar continuidade a trabalhos anteriores referentes à plataforma, tendo foco na sua manutenção, especificamente na documentação de requisitos e problemas existentes e na realização de testes.

Durante o decorrer do trabalho, o grupo documentou cerca de 40 problemas presentes na plataforma, detetados em trabalhos anteriores, por docentes e pelo grupo durante a sua análise. Adicionalmente, foi documentado um novo requisito que visou tornar a plataforma capaz de sustentar todo o processo de submissão e avaliação de trabalhos, tendo esta funcionalidade entrado em produção a tempo da entrega intermédia de 2º semestre da UC. Finalmente, foi implementada uma bateria de testes automáticos, que permitiu um acréscimo de 42 % da cobertura do código, passando de 45% para 87%, contribuindo assim para tornar a plataforma numa ferramenta mais estável e viável.

É de notar também que o grupo efetuou 3 inquéritos de satisfação conduzidos aos utilizadores da plataforma. Estes inquéritos forneceram *feedback* relativamente ao trabalho que estava a ser realizado, tornando-se imprescindíveis na componente de análise. Os temas abrangidos e os resultados encontram-se presentes neste relatório.

Abstract

The final course work is “a fundamental element of the curriculum of the ULHT Bachelor Degrees in LIG and LEI functioning as a core moment in the formation of students, namely in order to promote their professional integration and initiation to scientific work”.[1]

The management of this UC is done through a workflow management platform developed also for TFC, entitled “DEISI - TPC Management Platform” by Francisco Santos, initially developed in the 2017/2018 school year.

The platform boils down to a Spring MVC (Model-View-Controller) web application that allows “all stakeholders to be able to perform tasks in a fluid manner and respecting the normal workflow defined in the respective regulation.”.[2]

The present work aimed to continue previous works related to the platform, focusing on its maintenance, specifically in the documentation of existing requirements and problems and in the performance of tests.

During the course of the work, the group documented around 40 existing problems on the platform, detected in previous works, by teachers and by the group during its analysis. In addition, a new requirement was documented that aimed at making the platform capable of supporting the entire process of submitting and evaluating works, and this functionality went into production in time for the intermediate delivery of the 2nd semester of the UC. Finally, a battery of automatic tests was implemented, which allowed an increase of 42% in code coverage, going from 45% to 87%, thus contributing to make the platform a more stable and viable tool.

It should also be noted that the group carried out 3 satisfaction surveys conducted with users of the platform. These surveys provided feedback on the work that was being carried out, becoming essential in the analysis component. The topics covered and the results are included in this report.

1. Identificação do problema

a. Inquérito de satisfação a utilizadores

Inicialmente, de modo a obter *feedback* do estado da plataforma foi conduzido um inquérito de satisfação a um grupo de utilizadores através da plataforma *Google Forms*, entre os dias 3 e 17 de Novembro de 2019, no início do ano letivo.

O inquérito incluiu 11 perguntas cujas temáticas envolveram os temas: experiência de utilizador, interface de utilizador, funcionalidades, problemas encontrados e performance da plataforma. Foram fornecidas 18 respostas relevantes. Estas encontram-se em anexo neste documento.

b. Resultados

Em termos de **experiência de utilizador**, as respostas presentes nos gráficos 1 e 2, levam a crer que a grande maioria não enfrentou dificuldades em navegar pela plataforma. No entanto a **interface de utilizador**, tendo em conta as respostas presentes nos gráficos 3 e 4, não foi apontada como muito apelativa, podendo ser descrita como demasiado homogénea, visto que alguns pormenores no *design* gráfico foram considerados pouco chamativos.

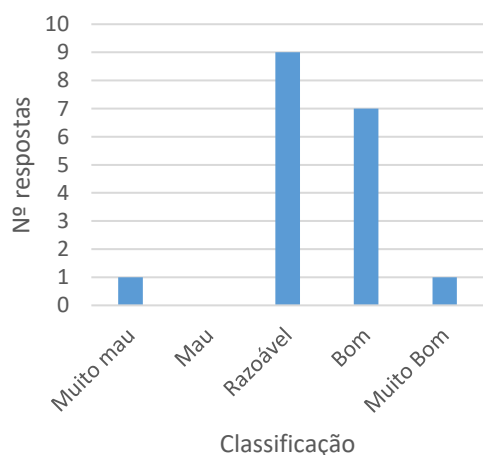


Gráfico 1 - Classifique o nível de intuição ao navegar pela plataforma.

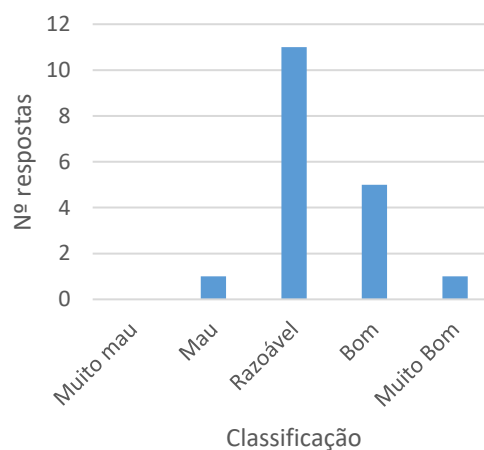


Gráfico 2 - Classifique o nível de organização e estruturação da plataforma.

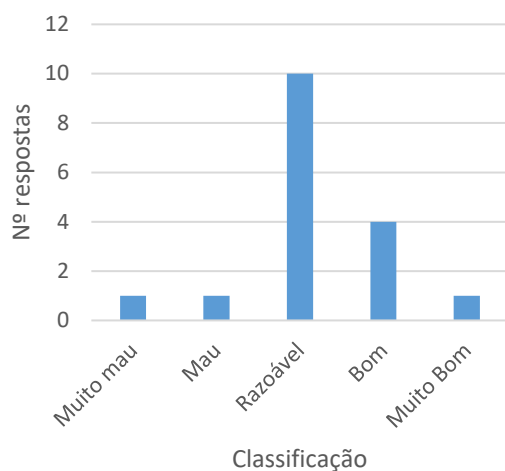


Gráfico 3 - Classifique o nível de consistência da interface.

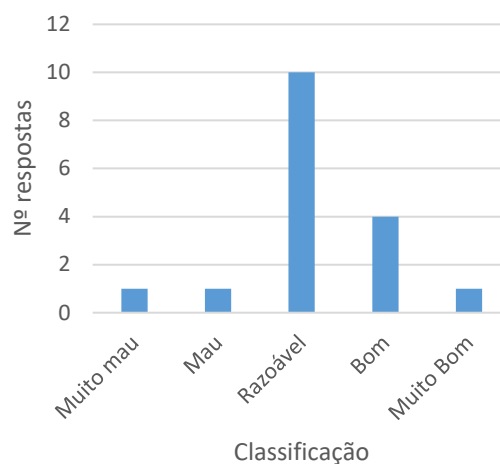


Gráfico 4 - Classifique o nível de apelação da interface.

Quanto à questão das **funcionalidades** fornecidas pela plataforma, o inquérito demonstra nos gráficos 5 e 6, a existência de utilizadores que se depararam com falhas durante a utilização da plataforma. No inquérito foi indicado um campo onde os questionados puderam listar tais impedimentos, entre os quais se destacaram:

- Impossibilidade de ver o conteúdo de alguns TFCs propostos por utilizadores do tipo discente;
- Caso o TFC ao qual um discente se candidatou fique reservado para outro discente, o mesmo não tem a possibilidade de cancelar a candidatura a esse tema, impedindo a realização de novas candidaturas;
- Durante a página de visualização de um TFC o botão de retroceder não funcionava.

É de notar que na altura em que o inquérito foi conduzido, encontravam-se descritos no repositório da plataforma um total de 28 falhas, entre as quais estavam as listadas anteriormente.

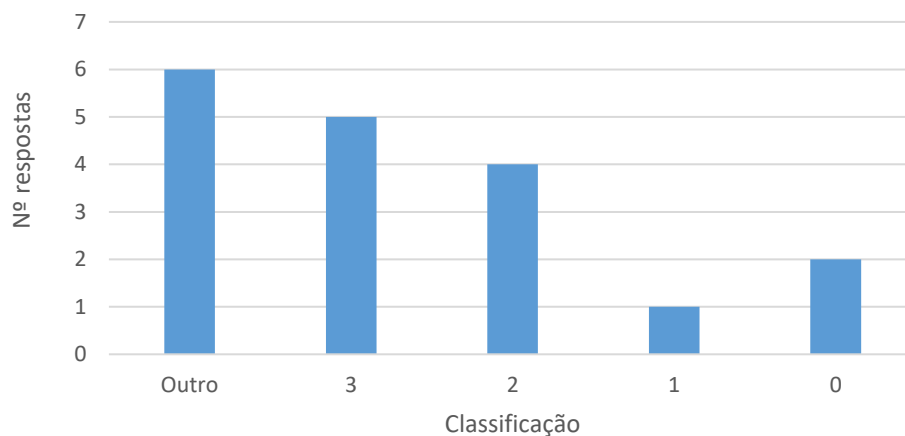


Gráfico 5 - Com quantos problemas se deparou ao utilizar a plataforma?

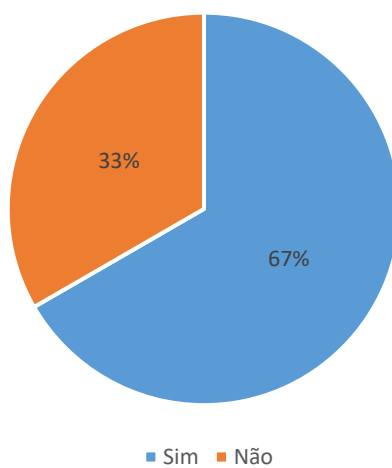


Gráfico 6 - Na sua opinião, a plataforma oferece todas as funcionalidades necessárias para o seu funcionamento ser adequado à UC?

Relativamente a questões de **performance**, de acordo com o gráfico 7, a grande maioria não apresentou queixa sobre nenhum impedimento ou inconveniente.

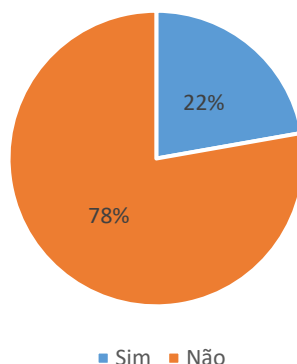


Gráfico 7 - Deparou-se com problemas de performance?

c. Conclusões

Na altura em que este inquérito foi realizado, a plataforma satisfazia apenas 9 dos 16 requisitos funcionais propostos na fase inicial do desenvolvimento do projeto. Estes requisitos satisfeitos, denominados de funcionalidades, proporcionaram aos seus utilizadores um conjunto de serviços mínimos que permitiram que a plataforma continuasse viável.

Tendo em conta as respostas do inquérito, denotou-se um nível de insatisfação por parte dos utilizadores, particularmente em termos de funcionalidades fornecidas.

O grupo concluiu que esta insatisfação remete indiretamente para o facto que, na altura em que o inquérito foi realizado, constavam apenas 7 testes sobre a plataforma, implicando uma baixa cobertura do código, inviabilizando a plataforma.

2. Viabilidade e pertinência

A gestão de TFCs inicialmente processava-se sem o apoio de nenhum sistema informático centralizado e integrado, sendo realizada através de ferramentas externas ao DEISI, como por exemplo *e-mails* e *google forms*.

A primeira implementação desta plataforma teve como objetivo “*dotar o DEISI de uma ferramenta que permita a realizar uma gestão desta Unidade Curricular*”.^[3]

Desde a sua primeira versão a plataforma já foi alvo de mais 2 projetos também realizados por outros discentes como tema de TFC em anos anteriores.

Para este trabalho, o grupo encarregou-se de efetuar uma análise aos problemas listados no repositório da plataforma, seguida da documentação e classificação por nível de prioridade dos mesmos. Esta classificação teve em conta o grau de impedimento apresentado, o nível de dificuldade e o tempo requerido para resolver o problema. De seguida, de modo a complementar a resolução dos problemas, o grupo procedeu à implementação de testes à plataforma.

Adicionalmente, a pedido da coordenação, o grupo documentou um novo requisito que ambiciona tornar a plataforma mais completa. Esta documentação, complementada depois por testes, foi então passada ao grupo de Bruno Bacelar e Rodrigo Marcelino, sobre o tema “Plataforma DEISI - Gestão de TFCs (Desenvolvimento)”, que procederam à implementação das novas funcionalidades.

a. Fatores económicos

Relativamente ao orçamento necessário para a realização do projeto, o grupo considera que não serão necessários fundos adicionais. Isto deve-se ao facto de apenas serem utilizadas ferramentas *open-source* gratuitas durante o seu desenvolvimento. Adicionalmente o grupo não apresenta nenhuma causa que indique que os recursos de *hardware* (memória, CPU, rede, entre outros) fornecidos pelo departamento que foram utilizados até à data sejam insuficientes.

3. Levantamento e análise de requisitos

Os requisitos referentes a este projeto foram inicialmente estabelecidos e documentados durante o ano letivo de 2017/2018 por Francisco Silva no seu TFC, que teve como objetivo desenvolver a plataforma. Como tal, os requisitos enumerados abaixo foram meramente consultados do relatório referente a essa primeira fase, apenas para efeitos de completude deste relatório.

Adicionalmente, novos requisitos estipulados durante esta fase de desenvolvimento encontram-se também listados nesta secção.

Segue-se a lista de requisitos inicialmente estabelecidos e implementados na plataforma.

a. Requisitos Funcionais

Requisito Funcional 1 (RF1) - Submissão de TFC Docente DEISI:

Após autenticação, os docentes DEISI devem conseguir submeter propostas de temas para TFC através de preenchimento de um formulário. Todas as propostas apresentadas pelos docentes devem de imediato passar ao estado de “À Espera de Escolha”. Todos os docentes que apresentem propostas, caso estas sejam escolhidas por um discente, são constituídos como Orientadores.

Requisito Funcional 2 (RF2) - Submissão de TFC Discente:

Todos os discentes necessitam autenticar-se de forma a aceder ao formulário de submissão de propostas, presente na sua área pessoal. Estes têm o direito de apresentar uma ou mais propostas (estas devem depois ser avaliadas pela Coordenação RF6). Todas as propostas apresentadas vão ficar no estado “A aguardar Aprovação”, até que sejam “Aprovados”, “Modificados” ou “Rejeitados” pela Coordenação da UC de TFC.

É de a responsabilidade da Coordenação atribuir um Orientador.

Requisito Funcional 3 (RF3) - Submissão de TFC Docente não DEISI:

Os docentes que não pertencem ao DEISI, não necessitam de se autenticar perante o sistema para submeter uma proposta de Tema de TFC. As propostas apresentadas irão ficar no estado “A aguardar Aprovação”.

Todas as propostas apresentadas necessitam de aprovação por parte da Coordenação (RF6) para passarem ao estado de Aceite.

Estes docentes devem assumir o papel de Co-orientador.

É de a responsabilidade da Coordenação atribuir um Orientador.

Requisito Funcional 4 (RF4) - Submissão de TFC Empresa/Entidade Externa:

Não existe a necessidade de autenticação perante o sistema.

Empresas e Entidades Externas devem indicar nas suas propostas um interlocutor, que irá assumir o papel de ligação com a instituição e de Co-orientador.

Todas as propostas para passarem para o estado de Aceite, necessitam de aprovação por parte da Coordenação (RF6).

É de a responsabilidade da Coordenação atribuir um Orientador.

Requisito Funcional 5 (RF5) - Criação de Grupo Trabalho:

O sistema deve permitir que os discentes decidam realizar o seu TFC em grupo (máximo dois elementos).

O grupo passa ao estado de aprovado, após ambos os alunos se autenticarem perante o sistema e confirmarem a aceitação de formar grupo. No momento de aprovação do grupo ambos os alunos devem receber uma notificação, via sistema, na sua área pessoal.

Requisito Funcional 6 (RF6) - Avaliação de TFCs:

É de a responsabilidade da Coordenação da UC de TFC avaliar as propostas apresentadas nos RF 2, 3 e 4.

Após aprovação estas propostas devem passar ao estado de “Em Espera de Escolha”, e deve ser enviada uma notificação ao autor da proposta (RF2 através do sistema, RF 3 e 4 via email).

Requisito Funcional 7 (RF7) - Candidatura a TFCs:

Todas as propostas apresentadas aceites (com a exceção das propostas do RF2) devem aparecer na *pool* de escolhas, de forma a que os discentes tenham acesso às mesmas. Estes necessitam se autenticar perante o sistema para aceder à *pool*. Devem conseguir escolher entre três a cinco propostas (por ordem de preferência). Podem candidatar-se individualmente ou em grupo. Caso seja em grupo, o grupo tem que estar previamente formado (RF5).

Requisito Funcional 8 (RF8) - Distribuição de Temas:

Após escolha por parte dos discentes dos temas (RF7), será de a responsabilidade da Coordenação atribuir os temas. O sistema deve fornecer toda a *pool* de escolhas realizadas (realçando situações em que múltiplos discentes tenham escolhido o mesmo tema).

A Coordenação para propostas dos RF 2, 3 e 4 deve conseguir atribuir um professor Orientador.

O docente/grupo que fique com um tema atribuído deve receber uma notificação, via sistema na sua área pessoal, a indicar que tema lhe foi atribuído.

Requisito Funcional 9 (RF9) - Criação de Júris e Agendamento de Apresentações:

O sistema deve permitir criar um conjunto de Júris para os dois momentos de avaliação (RF 10 e 11), assim que seja realizado um Agendamento (data definida pela Coordenação).

Todos os discentes que possam ser escolhidos para júri, devem receber uma notificação para que indiquem se dispõem de horário para fazer parte do mesmo.

A *pool* de júris para o Requisito 10 deverá ser composta pelos docentes que estejam a desempenhar o papel de Orientadores. Para o RF11 as *pools* de júris devem ser compostos no mínimo por dois e no máximo por três docentes (sendo obrigatório a presença do Orientador, e se existir um Co-orientador).

Estes júris devem ter acesso a todos os artefactos desenvolvidos pelo discente.

Os discentes (júri) e o discente/grupo devem ser notificados (via email e no sistema e no sistema, respetivamente) da data e hora para que ficou marcada a apresentação.

Requisito Funcional 10 (RF10) - *Dashboard* de acompanhamento Coordenação:

A Coordenação da UC de TFC deve conseguir acompanhar o estado em que se encontram os TFC a ser realizados, de forma a conseguirem acompanhar o desenvolvimento dos mesmos.

Requisito Funcional 11 (RF11) - Estado em que se encontra TFC:

O sistema deve permitir a identificação por parte do aluno em que situação se encontra o desenvolvimento do seu TFC, sendo que o Orientador deverá validar esta informação. Estados: Em TFC, Aceite, Rejeitado, A Melhorar, Atribuído, Avaliado em Avaliação Intermédia, Avaliado em Avaliação Final, Em Agendamento de Apresentação, Aceite para Avaliação e Rejeitado para Avaliação, Pendente entrega relatório intermédio; Pendente entrega relatório final.

Estados Em TFC, Aceite, Rejeitado, A Melhorar, Atribuído cabe à Coordenação atribuir. O Orientador deve definir os seguintes estados: Aceite para Avaliação e Rejeitado para Avaliação.

Requisito Funcional 12 (RF12) - Gravar Histórico do TFC:

O sistema deve criar um histórico da mudança de estados dos TFCs, guardando a data em que esta ocorreu bem como a informação do ator que o fez mudar (ex. quando um discente entrega os artefactos obrigatórios para a avaliação intermédia, TFC passa de Pendente Entrega Relatório Intermédio para Agendada Avaliação Intermédia, sistema deve registar a data da ocorrência da mudança e o número do discente).

Este histórico deve ser passível de consulta pelo discente e seu orientador, bem como por todos os membros da Coordenação.

Requisito Funcional 13 (RF13) - Configuração:

O sistema deve permitir à Coordenação conseguir configurar todos as datas/prazos relacionados com o TFC. Os prazos que devem estar presentes são: Período para Entrega de Temas de TFC (RF1 a 4); Data de Publicação de Temas RF1, 3 e 4; Período de Candidatura a TFC (RF7); Data de Publicação de Temas Atribuídos (RF8); Período de Avaliações definidas pela coordenação.

A Coordenação deve também conseguir inserir as tecnologias, que irão depois ser escolhidas em cada proposta de Tema de TFC.

b. Requisitos Não Funcionais

Requisito Não Funcional 1 (RNF1) - Usabilidade:

A ferramenta a ser desenvolvida deve ser simples, minimalista e de uso eficiente, de forma a se conseguir uma utilização agradável e que não crie entropias na sua utilização.

O sistema deve manter sempre o utilizador informado sobre o que se está a passar (ex. se estão autenticados ou não), a linguagem utilizada deve ser algo que os utilizadores consigam relacionar com o seu dia a dia, deve ser sempre oferecida uma solução para erros cometidos pelos utilizadores (ex. poderem voltar atrás com uma proposta de tema, caso tenham cometido um engano), todas as informações obrigatórias devem ser sinalizadas (de forma a facilitar o preenchimento por parte dos utilizadores) e sempre que exista um erro (ex. preenchimento errado de qualquer informação) a mensagem apresentada deve ser de fácil compreensão e apontar de imediato para onde se encontra o erro.

A plataforma deve ser desenvolvida para permitir ser visualizada em multiplataformas (permitir realizar operações tanto em desktop como em mobile).

Requisito Não Funcional 2 (RNF2) - Desempenho:

O sistema a ser desenvolvido necessita apresentar um bom desempenho, de forma a não provocar entraves ao bom funcionamento da mesma).

Visto a velocidade de internet disponível ser variável entre os diferentes utilizadores, a aplicação necessita ser leve (ex. imagens com um tamanho mais reduzido) de forma a que os tempos de *loading* das páginas sejam reduzidos (mesmo quando a velocidade disponível é reduzida).

Os acessos a elementos de base de dados necessitam ser realizados de forma rápida e o mais eficiente possível.

Requisito Não Funcional 3 (RNF3) - Segurança:

A autenticação de Docentes DEISI (e Coordenação da UC de TFC) e dos discentes deve ser integrada com o sistema Sign-On do estabelecimento de ensino.

Os docentes não DEISI e Empresas/Entidades Externas não necessitam de autenticação, sendo que o seu acesso ao portal desenvolvido deve-se restringir à parte onde irão submeter propostas para temas de TFC.

No caso dos docentes não DEISI, caso as suas propostas sejam aceites e escolhidas por algum discente, deverá ser atribuído um *username* e *password* para que estes possam acompanhar o trabalho desenvolvido (deve-lhes ser atribuído o papel de Coorientadores).

Todas as comunicações são estabelecidas através de um canal seguro (SSL).

Requisito Não Funcional 4 (RNF4) - Portabilidade:

Visto o sistema ir ser usado por um número alargado de utilizadores, este deve ser desenvolvido de forma a conseguir correr nos principais *browsers*: *Chrome*, *Firefox*, *Safari* e *Microsoft Edge*.

Requisito Não Funcional 5 (RNF5) - Ambiente Físico de Instalação:

A aplicação deverá ser instalada num servidor fornecido pela Universidade, com acesso pela Internet e um endereço IP público.

Segue-se agora a secção referente ao novo requisito estipulados neste trabalho:

Requisito Funcional 14 (RF14) - Avaliação de TFCs:

Os alunos devem fazer *upload* na plataforma dos relatórios e dos artefactos a entregar, até uma data de entrega, definida pela coordenação.

Após a submissão de todos os alunos, um elemento da coordenação despoleta a distribuição dos TFCs a docentes avaliadores. Todos estes docentes deverão ser do departamento DEISI.

Esta avaliação deverá ser submetida até uma data definida também pela coordenação. Opcionalmente, os docentes avaliadores podem indicar um número de TFCs que se disponibilizam a avaliar que o sistema deverá ter em conta durante a distribuição de TFCs.

De modo a proceder a uma avaliação, o docente avaliador deve aceder a uma lista de TFCs que o mesmo tem para avaliar. De seguida, deve seleccionar um trabalho para avaliar, onde, após analisar e ponderar sobre o mesmo, deverá indicar uma nota quantitativa no valor de 0 a 20 que deverá ser complementada de uma nota qualitativa em texto. Caso o momento de avaliação seja qualitativo, o docente avaliador não dará nota quantitativa.

Assim que as notas são lançadas, os alunos devem conseguir consultar as notas fornecidas pelos docentes avaliadores, em relação ao seu TFC.

As figuras 1, 2 e 3 representam os diagramas de casos de uso para este novo requisito.

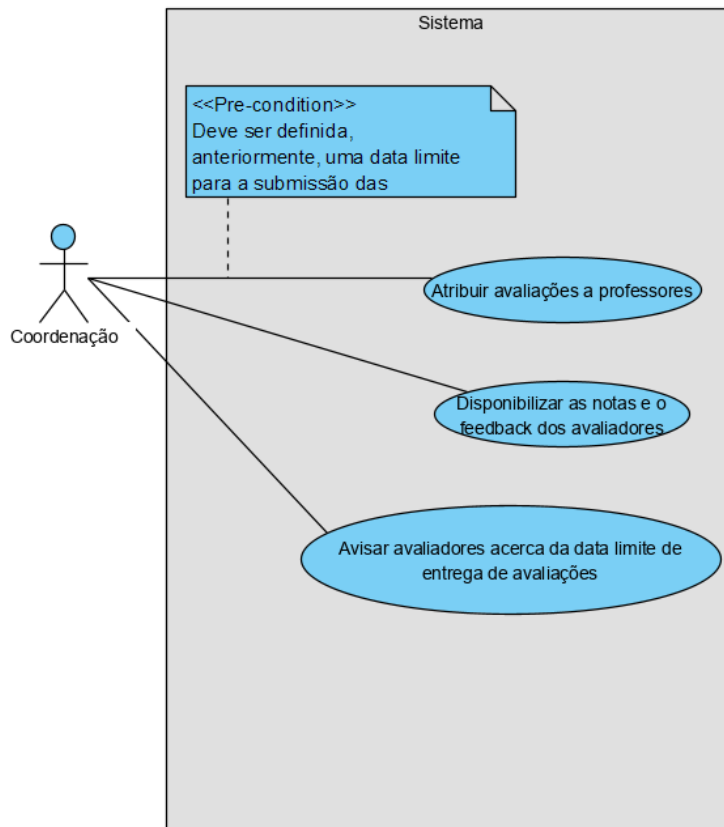


Figura 1 - Casos de uso do Coordenador.

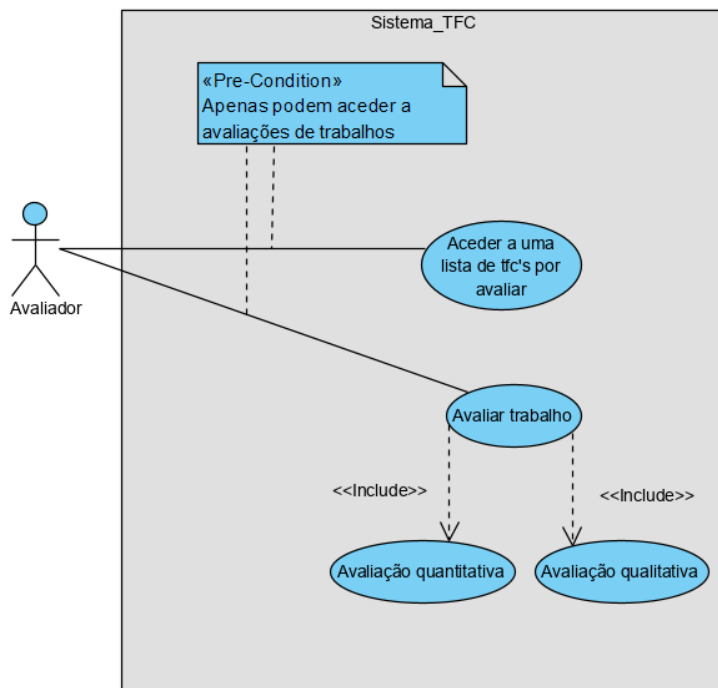


Figura 2 - Casos de uso do Docente Avaliador.

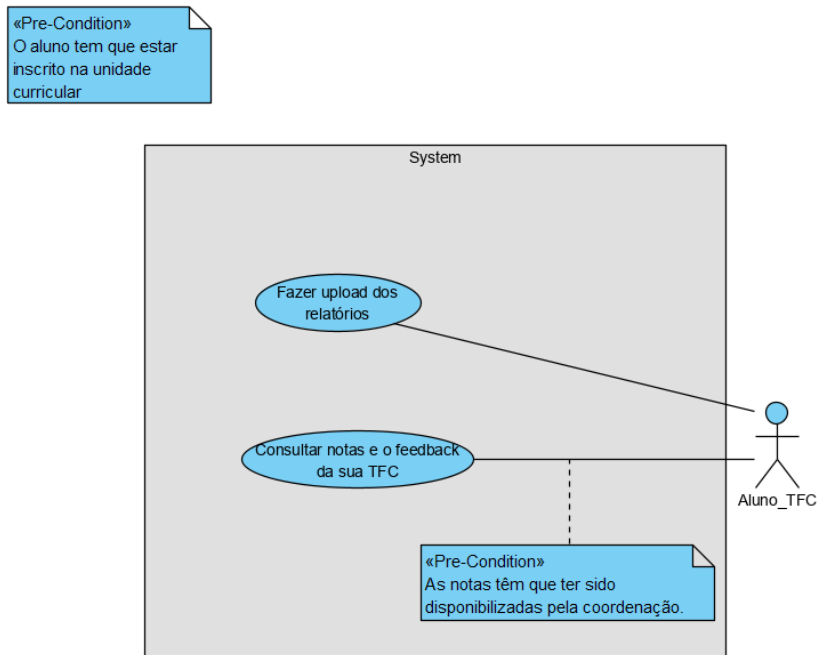


Figura 3 - Casos de uso do Aluno

A tabela 1 mostra o histórico de versões do requisito e a tabela 2 contém informação mais detalhada sobre o requisito.

Versão	Data	Autor(es)	Alterações efetuadas
1.0	27/12/2019	Rafael Horta Diogo César	Novo caso de uso
1.1	02/01/2020	Rafael Horta Diogo César	Adição de passos e de regras de processamento
1.2	07/12/2020	Rafael Horta Diogo César	Definição e alteração das regras de processamento de acordo com o <i>feedback</i> da coordenação

Tabela 1 - Histórico de versões: UC 14

ID e nome	UC - <i>Avaliação de TFC's</i>
Descrição	Neste caso de uso pretende-se que seja implementada na plataforma de tfc's, a possibilidade de avaliação dos trabalhos, de acordo com os requisitos abaixo explicados.
Papeis de utilizador	Aluno Professor DEISI Grupo
Fatores despoletantes	Através de um <i>click</i> na interface gráfica da aplicação
Pre-condições	1. O professor que realiza a atribuição de tfc's ao professores, de forma a serem avaliados, têm que ser da coordenação
Resultado final	1. As avaliações devem ser submetidas com sucesso, de forma os alunos verem a avaliação atribuída ao seu trabalho final de curso
Processo normal	<ol style="list-style-type: none"> 1. Um elemento da coordenação carrega num botão que permite que as tfc's sejam atribuídas a professores avaliadores. (KS1, KS2) <ol style="list-style-type: none"> 1.1 A coordenação tem que definir uma data para a entrega das avaliações (KS5) 2. De forma a efetuar a entrega dos reatórios, e dos artefactos, os alunos devem fazer upload no sistema do mesmo. (KS10) 3. Os avaliadores devem aceder a uma lista de tfc's que os mesmos devem avaliar. 4. Os avaliadores devem, de seguida, aceder ao respetivo trabalho que pretendem avaliar. (KS3, KS4, KS5) <ol style="list-style-type: none"> 4.1 Os avaliadores devem indicar a nota dada, justificando a nota dada, com base num campo que permita dar um <i>feedback</i> ao trabalho avaliado. (KS3, KS5, KS9) 5. Assim que as notas são lançadas, os alunos devem conseguir consultar a nota e o feedback dado pelo avaliador, em relação ao seu tfc. (KS6, KS7, KS8)
Processos opcionais	N/A
Regras de processamento	<p>KS1 Apenas um elemento da coordenação pode efetuar as atribuições dos trabalhos aos avaliadores.</p> <p>KS2 O Sistema deve distribuir os trabalhos a avaliar, de acordo com a carga de trabalho definida para cada avaliador, por parte da coordenação.</p> <p>KS3 Os avaliadores não podem avaliar trabalhos que estão atribuídos a outros avaliadores.</p>

	<p>KS4 Os avaliadores apenas podem aceder a avaliações de trabalhos por ele avaliados ou orientados.</p> <p>KS5 Os avaliadores têm que entregar as avaliações até à data definida pela coordenação</p> <p>KS6 Os alunos apenas podem consultar a nota e o <i>feedback</i> do seu respetivo trabalho.</p> <p>KS7 Os alunos não podem saber quem foram os avaliadores dos seus trabalhos.</p> <p>KS8 As avaliações devem ficar pendentes, até o coordenador indicar a sua publicação</p> <p>KS9 Se a avaliação for qualitativa, os júris não dão nota quantitativa, apenas inserem um comentário como <i>feedback</i>.</p> <p>KS10 Os elementos do júri devem poder aceder aos relatórios antes do período de avaliação.</p>
Dados	<p>T1 Aluno</p> <ul style="list-style-type: none">• Número de aluno• Nome• Curso• Email <p>T2 Grupo</p> <ul style="list-style-type: none">• ID do aluno 1• ID do aluno 2 <p>T3 Professor DEISI</p> <ul style="list-style-type: none">• Número do professor• Nome• Email <p>T4 TFC</p> <ul style="list-style-type: none">• ID do TFC• ID do Grupo/Aluno• Título• Descrição• Orientador• Coorientador (Caso exista)• Entidade (Caso exista) <p>T5 Avaliação (LT1)</p> <ul style="list-style-type: none">• ID do TFC avaliado• Número do professor atribuído• Nota atribuída

	<ul style="list-style-type: none"> • Observações
Exceções	N/A
Dados adicionais	LT1 A classe avaliação não existe no projeto existente, e portanto tem que ser criada e implementada.
Densidade de utilização	80 vezes por dia.

Tabela 2 - UC 14

4. Solução desenvolvida

a. Enquadramento prático

No livro “*Verification, Validation and Testing in Software Engineering*”[4], o autor classifica a **Engenharia de Testes** como um ramo da Engenharia de Software que tem como objetivo fornecer informações sobre a qualidade do software, relativamente ao contexto em que o mesmo deve operar, através dos conceitos de **verificação** e **validação**, que têm o propósito de expor as falhas de um sistema.

Enquanto a especificação do software diz respeito ao processo de verificação do software, a expectativa do cliente diz respeito ao processo de validação do software. Por meio da verificação será analisado se o produto foi feito corretamente e se ele está de acordo com os requisitos especificados. Por meio da validação será analisado se foi feito o produto correto e se ele está de acordo com as necessidades e expectativas do cliente.

Uma **falha** pode ser originada por diversos motivos, entre eles: a especificação de requisitos poder estar errada ou incompleta, ou a própria implementação pode estar incorreta, como um erro no algoritmo.

Para combater falhas são utilizados **testes**, que consistem num aglomerado de processos da Engenharia de Software que vão desde o **levantamento de requisitos** até à **execução do teste** propriamente dito, tendo como objetivo melhorar a qualidade do produto final. Estes testes podem ser vistos como uma parcela do processo de avaliação de qualidade de software.

De modo a priorizar a viabilidade da plataforma e garantir a pertinência do projeto, o principal foco do grupo será a execução de testes automáticos unitários e de *performance*.

A **cobertura de testes** e a **qualidade de código** são duas métricas fundamentais definidas na Engenharia de testes usadas para analisar, rastrear e medir a eficácia de um projeto. Isto porque uma cobertura de testes adequada ajuda a melhorar as hipóteses de detetar falhas críticas.

A cobertura de testes permite:

- Estimar a área de um requisito não implementada por uma bateria de testes;
- Identificar a quantidade de linhas de código testadas, o que diretamente confere para a qualidade do código;
- Identificar e criar cenários de teste adicionais que aumentam a cobertura;
- Identificar cenários de teste irrelevantes que não aumentam a cobertura;

Idealmente, uma bateria de testes deve cobrir entre 60% a 80% do código de uma aplicação. O cálculo deste valor pode ser efetuado da seguinte forma:

$$\text{cobertura de testes (\%)} = \frac{\text{n}^\circ \text{ de linhas cobridas}}{\text{n}^\circ \text{ total de linhas}} \times 100$$

Em conclusão, mensurar o bom funcionamento de um software envolve compará-lo com elementos como especificações, outros softwares da mesma linha, versões anteriores do mesmo produto, inferências pessoais, expectativas do cliente, normas relevantes, leis aplicáveis, entre outros. Para tal, deve ser escolhida a metodologia com a qual se pretende testar o software, tendo em conta os aspetos que se pretendem melhorar. Posteriormente devem ser escolhidos os tipos de testes que melhor se enquadram ao contexto dos problemas encontrados.

b. Tecnologias utilizadas

Como já mencionado anteriormente, na altura em que o inquérito de satisfação aos utilizadores foi conduzido, encontravam-se presentes na plataforma apenas 7 testes unitários. Estes testes foram criados com suporte à ferramenta *JUnit4*[6].

JUnit permite a criação e execução de testes para a tecnologia que serviu de suporte para a criação da plataforma - *Spring MVC*[7]. Esta trata-se de uma das mais conhecidas e utilizadas *frameworks* de *Java*. Implementa um grande número de funções, incluindo injeção de dependências, persistência de dados, entre outros. É de notar também que a ferramenta *Spring* torna a criação de testes usando *JUnit* muito mais simples, possuindo diversos suportes internos que permitem escrever casos de teste específicos para aplicações *web*.

De modo a lidar com a base de dados da plataforma é utilizado o *SGBD open-source MySQL*. Esta tecnologia traz múltiplas vantagens, entre elas segurança de dados, excelente desempenho e estabilidade, portabilidade, suporte transacional abrangente e excelente escalabilidade.

Adicionalmente, o *IDE IntelliJ IDEA*, será utilizado para o desenvolvimento do projeto, pois este tem suporte para todas estas ferramentas.

c. Ambiente de desenvolvimento

i. Análise dos problemas documentados no repositório da plataforma

Após ter sido conduzido o inquérito de satisfação referente ao estado da plataforma, o grupo começou por realizar uma análise mais aprofundada à causa dos problemas descritos. Esta análise teve em conta os requisitos propostos inicialmente pelo aluno que deu início ao desenvolvimento da plataforma (que foram vistos na secção anterior) e os problemas documentados da plataforma, disponíveis num repositório privado no *GitHub*.

O grupo procedeu à documentação destes problemas de forma mais profunda, indicando o procedimento para obter o problema, o resultado obtido, o resultado esperado e o nível de impedimento apresentado pelo problema.

Na figura 4, pode-se verificar parte da documentação feita no repositório. Uma documentação mais completa encontra-se nos anexos deste relatório.

34 Open	3 Closed	Author	Label	Projects	Milestones	Assignee	Sort
1							
1							
1							
1							
3							
3							
1							
1							
3							
3							
1							
1							
3							
3							
1							

Figura 4 - Listagem de problemas no repositório da plataforma.

Na figura 5, encontra-se um exemplo de um comentário criado pelo professor Pedro Alves, neste caso em concreto, referente ao problema “Rever o fluxo da proposta entre a submissão e a aprovação”.

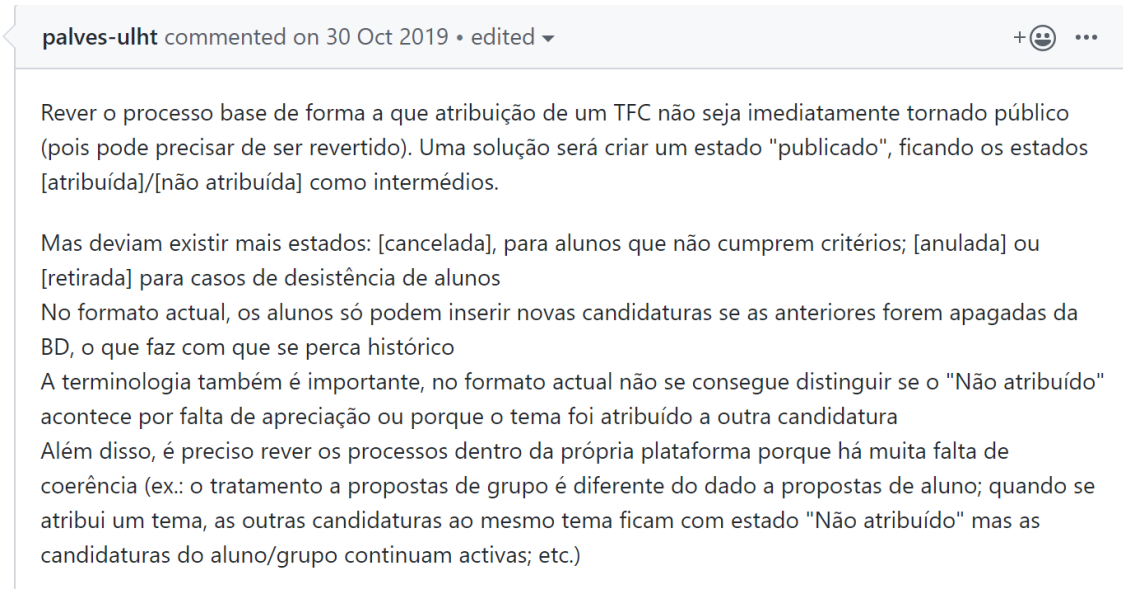


Figura 5 - Comentário do professor Pedro Alves que deu a conhecer o problema.

Após a análise deste problema exposto pelo professor, o grupo procedeu à documentação mais detalhada do mesmo. No entanto, a proposta foi refutada pelo outro grupo, referente ao TFC – Plataforma DEISI – Gestão de TFCs (Desenvolvimento) - indicando que faltava indicar um estado referente ao TFC. Após reanalisar o problema, o grupo corrigiu a proposta para a resolução do mesmo. Na figura 6, encontra-se o comentário referente à correção.

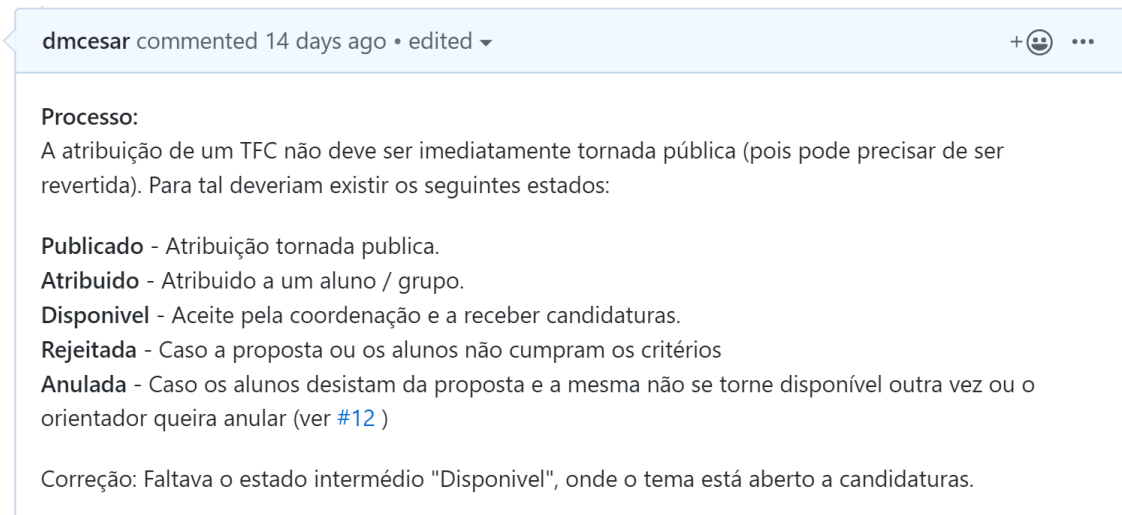


Figura 6 - Proposta de resolução indicada pelo grupo após a correção.

Está disponível em anexo neste documento, a tabela 1 que relaciona o ID dos problemas listados no repositório, com o seu nome, o tipo (*Enhancement* ou *Bug*) e o seu nível de impedimento, para consulta futura.

ii. Escrita de testes automáticos

Um dos principais focos deste trabalho foi a validação e verificação da plataforma através do uso de testes automáticos. Estes testes foram escritos maioritariamente em ambiente de desenvolvimento, ou seja, localmente e com quaisquer alterações a terem efeito no momento. Esta flexibilidade permitiu ao grupo subir o valor da cobertura de testes de um valor de 45% para 86%. Durante esta fase foram encontrados alguns problemas que foram posteriormente documentados no repositório da plataforma.

d. Ambiente de produção e *Test-Driven Development*

Durante o decorrer deste trabalho, o grupo documentou um novo requisito que visa tornar a plataforma capaz de sustentar a submissão e, posteriormente, a avaliação de artefactos submetidos por alunos inscritos na UC (a documentação detalhada deste requisito pode ser consultada no próprio documento, no capítulo 3).

As funcionalidades documentadas neste novo requisito foram implementadas em *Test-Driven Development* (Desenvolvimento Orientado a Testes ou *TDD*). Esta técnica consiste na especificação do software baseada em testes onde estes são escritos quase em simultâneo à implementação das funcionalidades. Desta forma, o programador tem acesso a *feedback* instantâneo e a uma especificação mais detalhada, mitigando assim a probabilidade de erros. O uso desta técnica foi um desafio para ambos os grupos, pois exigiu uma constante coordenação de ambos os lados que incluiu a escrita e correção de novos testes e a descoberta e documentação de novos problemas.

Durante este período, o grupo descobriu 18 falhas na implementação das novas funcionalidades graças à criação de novos testes, 9 das quais já foram corrigidas.

A figura 7 demonstra a lista de falhas detetadas pelo grupo que ainda não foram corrigidas e a figura 8 demonstra a lista de falhas detetadas pelo grupo e, posteriormente, corrigidas pelo grupo de desenvolvimento.

<input type="checkbox"/>	11 Open	9 Closed	Author	Label	Projects	Milestones	Assignee	Sort
<input type="checkbox"/>	🔔			Escolha da época de avaliação ao visualizar avaliações	enhancement			
				#22 opened yesterday by BrunoBacelar				
<input type="checkbox"/>	🔔			Erro ao editar propostas	bug			
				#21 opened 2 days ago by dmcesar				
<input type="checkbox"/>	🔔			Melhorias no histórico	enhancement			
				#20 opened 2 days ago by a21705375				
<input type="checkbox"/>	🔔			Melhoria nas entradas do historico	enhancement			
				#19 opened 2 days ago by dmcesar				
<input type="checkbox"/>	🔔			Melhoria de UI no ecrã /atribuir tfc	enhancement			
				#18 opened 2 days ago by dmcesar				
<input type="checkbox"/>	🔔			Criação de época de avaliação antes de criação de temas impede a atribuição de juris	bug			
				#17 opened 2 days ago by dmcesar				
<input type="checkbox"/>	🔔			Os alunos podem re-submeter trabalhos mesmo dps dos juri os avaliarem	bug			
				#16 opened 3 days ago by a21705375				
<input type="checkbox"/>	🔔			Não há validação de hora a submeter épocaAvaliacaoForm	bug			
				#15 opened 3 days ago by a21705375				
<input type="checkbox"/>	🔔			Re-atribuição de tema causa perda de tema	bug			
				#14 opened 3 days ago by dmcesar				

Figura 7 - Falhas detetadas e ainda não corrigidas.

<input type="checkbox"/>	🔔			De momento, é possível re-publicar as avaliações como coordenador	bug			1
				#13 by dmcesar was closed yesterday				
<input type="checkbox"/>	🔔			Atribuição do mesmo júri multiplas vezes para o mesmo trabalho	bug			
				#12 by dmcesar was closed yesterday				
<input type="checkbox"/>	🔔			Erro ao fazer download de um relatorio durante avaliação	wontfix			2
				#11 by dmcesar was closed 2 days ago				
<input type="checkbox"/>	🔔			Ao atribuir os juris, se fizer reload à página o primeiro jurado (orientador) desaparece				1
				#10 by dmcesar was closed 2 days ago				
<input type="checkbox"/>	🔔			Erro ao submeter avaliação depois de editar a mesma				1
				#8 by a21705375 was closed 3 days ago				
<input type="checkbox"/>	🔔			Um júri consegue avaliar uma TFC, que não é suposto avaliar				1
				#7 by a21705375 was closed 3 days ago				
<input type="checkbox"/>	🔔			Criação de época de avaliação para data expirada	bug			1
				#6 by dmcesar was closed 3 days ago				
<input type="checkbox"/>	🔔			Um aluno consegue aceder à página de avaliações				5
				#4 by a21705375 was closed 2 days ago				
<input type="checkbox"/>	🔔			Erro ao submeter um form de época de avaliação				
				#3 by a21705375 was closed 5 days ago				

Figura 8 - Falhas detetadas e corrigidas durante período TDD.

Como etapa final, foi ambicionada a colocação da plataforma em ambiente de produção até dia 26/04/2020, a tempo da entrega intercalar do 2º semestre da UC. Desta forma, todos os alunos podem submeter os seus artefactos (relatórios, vídeos, protótipos, etc.) num ambiente que permite também aos júris atribuir diretamente uma avaliação. O repositório de testes encontra-se no seguinte endereço: <https://github.com/dmcesar/gestao-tfcs>.

5. Benchmarking

a. Análise comparativa

Uma das principais alternativas à *framework JUnit*, é a *framework TestNG*[8]. Ambas são *open-source* e as suas funcionalidades são praticamente as mesmas, com a única exceção que JUnit não oferece testes de dependências. No entanto, o grupo optou por utilizar JUnit pois existe há mais tempo, o que contribui para um melhor suporte de comunidade, estando definido como o padrão para testes unitários em aplicações baseadas em *Java*.

Outra possibilidade seria o uso das *frameworks JBehave*[9] e *Serenity*[10] que incidem sobre *Behavior Driven Development* (BDD). O principal objetivo destas ferramentas é tornar a fase de testes de uma aplicação mais baseada no seu comportamento. O grupo decidiu não utilizar esta abordagem, pois esta foca-se muito no conceito de validação de software, onde se destaca a colaboração entre *stakeholders* e se minimizam os detalhes técnicos.

6. Plano de testes e validação

Neste capítulo, encontram-se descritos todos os testes automáticos escritos pelo grupo. Os testes têm como objetivos: aumentar a cobertura do código, complementar a correção de problemas analisados previamente e testar novas funcionalidades.

1. testLogin()

Descrição: Este teste é efetuado um pedido *HTTP GET* para o *URL /login*. Como resultado, o utilizador deve ser redirecionado para a página de login.

2. testLoginELogout()

Descrição: Este teste é uma extensão do teste 2. com a adição em que é adicionado um pedido *HTTP POST* para o *URL /logout*.

3. testDocenteInserePropostaEAlunoInscreve()

Descrição: Este teste põe em prática todo o processo de associação de um tema a um aluno. Inclui a criação do tema por parte de um docente, a candidatura de um aluno ao tema em questão e a atribuição do tema ao aluno por parte da coordenação.

4. testDocenteInserePropostaEGrupoInscreve()

Descrição: Este teste é similar ao teste 1. sendo que a única diferença se situa no processo de candidatura que, neste caso, menciona um segundo aluno, ou seja, submete uma candidatura em grupo.

5. testFormNDEISI()

Descrição: Este teste especifica o processo de criação de um tema, recorrendo ao uso de um formulário destinado para docentes que não pertencem ao departamento DEISI.

6. testFormEmpresa()

Descrição: Este teste é similar ao teste 5., especificando o processo de criação de um tema sendo que, neste caso é utilizado um formulário destinado para empresas/entidades externas.

7. testFormDocenteIncompleto()

Descrição: Este teste visa expor uma mensagem de erro que alerta o utilizador quando este submete um formulário destinado à criação de propostas com campos mal preenchidos ou não preenchidos.

8. testCoordenadorAceitaPropostaComErros()

Descrição: Este teste valida tanto a atribuição de um tema criado por um aluno a ele próprio como a mensagem de erro do formulário de aceitação de propostas de alunos, mostrada ao submeter o formulário com campos mal preenchidos ou não preenchidos.

9. testInsercaoEDeleteProposta()

Descrição: Este teste valida a consistência do repositório de dados ao remover uma proposta previamente criada por um docente.

10. testInscricaoEDelete()

Descrição: Este teste é uma variante do teste 9. onde em vez de proceder à remoção de uma proposta, remove uma inscrição ao tema por parte de um aluno.

11. testDeletePropostaInexistente()

Descrição: Este teste é outra variante do teste 9. no sentido em que procede à remoção de uma proposta com ID inexistente, recorrendo a um pedido *HTTP GET* para o *URL /delete/{id_proposta}*.

12. testConsistenciaHistorico()

Descrição: Este teste valida a consistência do histórico apresentado do repositório de dados após a inserção de uma nova proposta por parte de um aluno.

13. testCoordenacaoAceitaProposta()

Descrição: Este teste é uma variante do teste 8. onde um coordenador aceita uma proposta criada por um aluno. De seguida, é validado se a proposta se apresenta como atribuída ao aluno e se esta se apresenta também na sua lista de propostas.

14. testDashboardEmpresa()

Descrição: Este teste valida se o acesso ao *dashboard* destinado para empresas / entidades externas é acessível.

15. testDashboardNDEISI()

Descrição: Este teste valida se o acesso ao *dashboard* destinado para docentes que não pertencem ao departamento DEISI é acessível.

16. testEditarProposta()

Descrição: Este teste valida a consistência do repositório de dados ao atualizar uma proposta previamente criada por um docente.

17. testInscricaoAluno()

Descrição: Este teste valida todo o processo de candidatura de um aluno para um tema, incluído a criação do tema por parte do docente, a visualização do tema na lista de temas disponíveis e a visualização em detalhe do tema por parte do aluno.

18. testFormNDEISIIncompleto()

Descrição: Este teste valida se é apresentada uma mensagem de erro ao preencher incorretamente um formulário de criação/edição de temas destinado para docentes que não pertencem ao departamento DEISI.

19. testFormEmpresaIncompleto()

Descrição: Este teste valida se é apresentada uma mensagem de erro ao preencher incorretamente um formulário de criação/edição de temas destinado para empresas / entidades externas.

20. testAcederFormAluno()

Descrição: Este teste valida o acesso ao formulário de criação de temas destinado para alunos.

21. testAcederHistorico()

Descrição: Este teste valida o acesso ao ecrã que apresenta o histórico de dados contido no repositório de dados, destinado a docentes/coordenadores.

22. testFormAlunoIncompleto()

Descrição: Este teste valida se é apresentada uma mensagem de erro ao preencher incorretamente um formulário de criação/edição de temas destinado para alunos.

23. testVerInscricoes()

Descrição: Este teste realiza a criação de várias propostas e, posteriormente, a candidatura de alunos às mesmas. Como finalidade, o coordenador deve poder ver listadas as inscrições dos alunos às propostas.

24. testAtribuicaoMalFeita()

Descrição: Este teste valida se é apresentada uma mensagem de erro ao preencher incorretamente um formulário de atribuição de tema.

25. testAtribuiTFC()

Descrição: Este teste valida todo o processo de atribuição de um tema a um aluno, incluindo a criação do tema, a candidatura e a atribuição em si.

26. testEditarTFCEmpresa()

Descrição: Este teste valida a consistência do repositório de dados ao editar um TFC criado com recurso ao formulário destinado a empresas / entidades externas.

27. testEditarTFCNDEISI()

Descrição: Este teste valida a consistência do repositório de dados ao editar um TFC criado com recurso ao formulário destinado a docentes que não pertençam ao departamento DEISI.

28. testInscricaoComNumeroAlunoRepetido()

Descrição: Este teste valida a impossibilidade de submeter uma candidatura em grupo onde os números de aluno indicados são iguais.

29. testInserirTecnologiaVazia()

Descrição: Este teste valida a impossibilidade de inserir uma tecnologia vazia (“ ”) no campo de seleção de tecnologias ao criar uma nova proposta.

30. testInscricaoNumeroAlunoInvalido()

Descrição: Este teste valida a impossibilidade de indicar um número de aluno com formato inválido no campo de seleção do número do 2º aluno num formulário de candidatura em grupo.

31. testSubmeterFormAlunoComoDocente()

Descrição: Este teste valida a impossibilidade de submeter um formulário de criação de propostas destinado para alunos como docente.

32. testSubmeterFormDocenteComoAluno()

Descrição: Este teste valida a impossibilidade de submeter um formulário de criação de propostas destinado para docentes como aluno.

33. testSubmeterFormNDEISIComoAluno()

Descrição: Este teste valida a impossibilidade de submeter um formulário de criação de propostas destinado para docentes que não pertencem ao departamento como aluno.

34. testSubmeterFormEmpresaComoAluno()

Descrição: Este teste valida a impossibilidade de submeter um formulário de criação de propostas destinado para empresas / entidades externas como aluno.

35. testProcessoAvaliacaoQuantitativaTFC()

Descrição: Teste onde docente submete proposta, um aluno se inscreve e um coordenador atribui. De seguida, o coordenador cria uma época de avaliação qualitativa, durante a qual o aluno submete o seu relatório. De modo a possibilitar a avaliação, o coordenador atribui 2 jurados (para além do orientador) cuja função é avaliar o trabalho. Após esta função ser cumprida os jurados verão que não têm mais trabalhos por avaliar e, tanto o aluno como o orientador poderão verificar os comentários atribuídos pelos jurados.

36. testProcessoAvaliacaoQuantitativaTFC()

Descrição: Teste onde docente submete proposta, um aluno se inscreve e um coordenador atribui. De seguida, o coordenador cria uma época de avaliação quantitativa, durante a qual o aluno submete o seu relatório. De modo a possibilitar a avaliação, o coordenador atribui 2 jurados (para além do orientador) cuja função é avaliar o trabalho. Após esta função ser cumprida os jurados verão que não têm mais trabalhos por avaliar e, tanto o aluno como o orientador poderão verificar as notas atribuídos pelos jurados.

37. testAcederEcraParaAvaliarTrabalhos()

Descrição: Teste que valida o acesso à página onde um júri seleciona trabalhos para avaliar.

38. testAcederAvaliacaoTFCInexistente()

Descrição: Teste que visa validar a *error page* mostrada quando se tenta aceder à avaliação de um TFC inexistente ao realizar um pedido *HTTP GET*.

39. testJuriTentaAvaliarTrabalhoQueNaoLheEstaAtribuido()

Descrição: Teste onde um docente submete uma proposta, um aluno se inscreve e um coordenador atribui. De seguida, o coordenador cria uma época de avaliação, durante a qual o aluno submete o seu relatório. Posteriormente, um júri que não foi atribuído para avaliar o trabalho tenta aceder ao mesmo. O mesmo tem acesso à página mas ao tentar submeter uma avaliação receberá uma *error page* com o código 401 (*forbidden*).

40. testAlunoTentaAcederListaAvaliacoesSemAutorizacao()

Descrição: Teste onde um aluno tenta aceder à página que contém todas as avaliações atribuídas.

41. testAlunoTentaPublicarAvaliacoes()

Descrição: Teste onde um aluno tenta submeter as avaliações (tarefa do coordenador) através de um pedido *HTTP POST*.

42. testAcederEcraProibido()

Descrição: Teste onde um aluno tenta aceder ao ecrã ao qual é redirecionado quando tenta executar uma ação à qual não tem permissões para realizar.

43. testFormEpocaAvaliacaoInvalido()

Descrição: Teste onde um coordenador tenta submeter um formulário de criação de uma época de avaliação indevidamente preenchido. Como resultado, o coordenador deverá permanecer na página, onde lhe são indicados os campos mal preenchidos.

44. testFormComDataInvalida()

Descrição: Teste que aprofunda os casos em que a submissão de um formulário de criação de época de avaliação é inválido, inserindo uma data limite inválida (anterior à data atual do sistema).

45. testValidarCoerenciaInsercaoEpocaAvaliacao()

Descrição: Teste que valida a coerência da inserção de uma nova época de avaliação, ou seja, se depois da submissão de um formulário devidamente preenchido, se a época de avaliação se encontra listada e em efeito.

46. testAlunoEditaEpocaAvaliacao()

Descrição: Teste que verifica as permissões de um aluno que tenta editar uma época de avaliação. O resultado deverá ser uma *error page* que alerta para a falta de permissões para a ação.

47. testAtualizarEpocaAvaliacao()

Descrição: Teste que valida a coerência dos dados de uma época de avaliação que sofre alterações.

48. testApagarEpocaAvaliacao()

Descrição: Teste que verifica a coerência do tamanho da lista que contém as épocas de avaliação após a remoção de uma época de avaliação.

49. testDownloadFile()

Descrição: Teste que verifica a integridade de um ficheiro que foi submetido por um aluno e, posteriormente, requisitado por um júri. O tamanho do ficheiro obtido pelo júri deve ser igual ao tamanho do ficheiro submetido pelo aluno.

50. testDownloadFicheiroInexistente()

Descrição: Teste onde um utilizador tenta obter um ficheiro inexistente com recurso a um pedido *HTTP GET*.

51. testDownloadSemPermissão()

Descrição: Teste onde um aluno tenta obter o ficheiro submetido por outro aluno com recurso a um pedido *HTTP GET*.

52. testGrupoSubmeteUpload()

Descrição: Teste que valida o upload de um ficheiro por parte de um grupo. Após o upload, o utilizador deve ser redirecionado e deve ser apresentada uma mensagem de sucesso. Adicionalmente, o ficheiro deve ficar armazenado na base de dados.

53. testUploadEmptyFile()

Descrição: Teste onde um aluno realiza o upload de um ficheiro inexistente.

54. testUploadFile()

Descrição: Teste onde um aluno tenta realizar o upload de um ficheiro durante uma época de avaliação.

55. testUploadSemFicheiro()

Descrição: Teste onde um aluno submete o formulário de submissão de trabalho sem realizar upload de um ficheiro com recurso a um pedido *HTTP POST*.

56. testAcederEcraUpload()

Descrição: Teste que valida o acesso ao ecrã de submissão de ficheiros para avaliação.

57. testSubmeterFormComJuriAMenos()

Descrição: Para este teste, o coordenador cria uma proposta a que depois atribui a uma inscrição válida de um aluno. Posteriormente, o coordenador realiza a atribuição de júris a essa proposta com um elemento a menos do obrigatório. O utilizador deve ser redirecionado para a página com uma mensagem de erro.

58. testSubmeterFormComErros()

Descrição: Para este teste, é executado um pedido para a página */juriForm* onde não foi passado como argumento nem o ID do TFC nem a atribuição de nenhum júri.

59. testFluxoAtribuicaoJuri()

Descrição: Neste teste, é submetida uma proposta a que depois é atribuída a um aluno. De seguida, o coordenador cria uma época de avaliação durante a qual o aluno submete um artefacto. Depois, de modo a realizar a avaliação do trabalho, o coordenador navega para a lista de trabalhos a atribuir júri onde seleciona o trabalho e submete o formulário com os 3 júri obrigatórios selecionados. Finalmente, os 3 júris que foram atribuídos ao trabalho podem ser visualizados na lista de trabalhos a avaliar, associados ao trabalho.

60. testSubmeterFormSemAutorizacao()

Descrição: Teste que valida se a plataforma não permite a submissão do formulário de atribuição de júris por parte de utilizadores sem permissões.

61. testPublicarAvaliacoesSemAutorizacao()

Descrição: Teste que valida se a plataforma não permite a publicação da atribuição de júris por parte de utilizadores sem permissões.

62. testPostJuriForm()

Descrição: Este teste valida o fluxo de atribuição de júri para um trabalho submetido por um aluno.

63. testJuriFormInexistente()

Descrição: Neste teste, é validado o erro que deve ocorrer ao tentar submeter um formulário de atribuição de júri para um trabalho com ID não atribuído.

64. testAcederEcraJuri()

Descrição: Este teste valida o acesso à página que lista os trabalhos a avaliar para cada júri.

65. testPublicarListaJurisVazia()

Descrição: Este teste engloba a criação e atribuição de uma proposta a um aluno com a atribuição de um júri e a sua avaliação. Após a atribuição do júri ao trabalho, o mesmo visualiza o trabalho e atribui uma nota, preenchendo um formulário. Com efeito após a submissão do formulário, tanto o aluno como o coordenador têm acesso à avaliação atribuída. Adicionalmente, o júri deixa de visualizar o trabalho na sua lista de trabalhos por avaliar, indicando que o mesmo já foi submetido.

66. testJuriFormComJurisIguais()

Descrição: Neste teste, um coordenador tenta submeter o formulário de atribuição de júris a um trabalho com dois ou mais júris com o mesmo nome.

7. Resultados

Durante o decorrer do trabalho, o grupo contribuiu para a viabilização da plataforma, realizando uma análise à plataforma que resultou na documentação de mais de 50 problemas descritos no repositório da mesma.

Adicionalmente, o grupo criou uma bateria de testes, recorrendo à *framework JUnit 4*, que permitiu um aumento da cobertura do código de 45% para 86%, mitigando o risco de novas falhas serem encontradas.

Finalmente, foi documentado e também testado (após a sua implementação), um novo requisito que permitiu à plataforma passar a englobar todo o processo de submissão e avaliação de trabalhos. É de notar que este requisito chegou a ser incluído em ambiente de produção a tempo da entrega relativa à avaliação intercalar de 2º semestre da UC.

De modo a receber *feedback* dos utilizadores da plataforma, o grupo realizou dois inquéritos, quer a alunos, quer a docentes dos vários perfis (júri, orientador e coordenador), ambos focados na experiência dos questionados ao utilizarem a plataforma durante o processo de submissão e avaliação de trabalhos.

a. Inquérito aos alunos

O primeiro inquérito, destinado para alunos, teve como objetivo receber feedback sobre todo o processo de submissão de trabalhos na plataforma.

O inquérito recebeu 12 respostas, as quais o grupo acredita que foram suficientes para que fosse possível extrapolar conclusões.

O objetivo para este inquérito foi descobrir eventuais erros no processo de submissão de trabalhos e melhorias que os utilizadores gostariam de ver na plataforma.

i. Resultados

Tendo em conta as respostas obtidas, nenhum utilizador se deparou com nenhum problema ao utilizar a plataforma durante a sua experiência ao realizar a submissão do seu trabalho.

Finalmente, foi perguntado aos utilizadores que melhorias ou novas funcionalidades gostariam de ver implementadas na plataforma. Sendo esta pergunta opcional, apenas foram obtidas duas respostas. Ambas as respostas apontaram para uma eventual confirmação de que um trabalho foi submetido. Sendo que a plataforma já realiza este feito, o grupo criou a hipótese que o problema apontado por ambas as respostas, tenha sido resultado de falta de atenção dos utilizadores. No entanto, o grupo acredita que estas respostas não devem ser descartadas e que uma melhoria pode ser efetuada. Uma possível solução para este problema, poderá ser a futura implementação de um sistema que notifique os utilizadores via *email*, sempre que um trabalho é submetido. Adicionalmente, poderá ser facultada ao utilizador, uma opção na plataforma que permita ao mesmo realizar o *download* do último artefacto que submeteu.

b. Inquérito aos docentes

O segundo inquérito, destinado para docentes dos vários perfis, teve como objetivo receber feedback sobre todo o processo de avaliação de trabalhos na plataforma, incluindo a criação de novas épocas de avaliação, a atribuição de júris a trabalhos submetidos, a obtenção e avaliação de trabalhos por parte dos júris, a visualização de avaliações atribuídas por parte de orientadores e a revisão e publicação de avaliações atribuídas por parte de coordenadores.

Este inquérito obteve 2 respostas, o que o grupo considera um número inferior ao pretendido. No entanto, foram obtidas respostas relevantes.

i. Resultados

O inquérito começa pelo perfil de avaliador do questionado. Segundo as respostas obtidas, todos os docentes conseguiram visualizar os trabalhos que lhes foram atribuídos para avaliar, sendo também capazes de submeter as suas avaliações relativamente a esses trabalhos.

De acordo com o gráfico 8, um dos questionados não conseguiu realizar o download de um relatório. No entanto, na pergunta seguinte do inquérito, onde os questionados são convidados a deixar sugestões na plataforma relacionadas com o seu perfil, o docente que respondeu que não conseguiu realizar o download, afirmou que após reportar o problema em questão, o mesmo foi “*rapidamente corrigido*”.

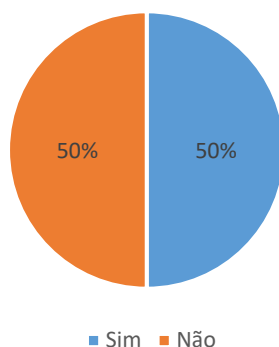


Gráfico 8 - Conseguiu realizar o download dos trabalhos que lhe foram atribuídos?

Finalmente, ainda no perfil de avaliador, foi indicada outra sugestão, onde um docente gostaria de ser notificado via *email* quando a lista de trabalhos que lhe são atribuídos é publicada.

Passando para o perfil de orientador, de acordo com o gráfico 9, um dos questionados afirma que não conseguiu visualizar as avaliações relativamente aos trabalhos que orienta. O grupo acredita que este problema se deve ao facto de que os trabalhos de um orientador apenas são visíveis após as respetivas avaliações serem atribuídas. Como abordagem para este problema, o grupo sugere que deverão passar a ser listados todos os trabalhos orientados por um docente na página que cobre o seu perfil de orientador. Adicionalmente, os trabalhos que ainda não foram avaliados deverão indicar um estado que represente que a avaliação se encontra pendente.

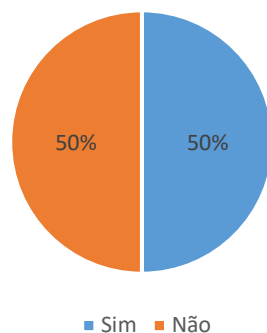


Gráfico 9 - Conseguiu listar e ver as avaliações aos trabalhos que orienta, à medida que estas foram submetidas?

Para terminar as observações ainda sobre o perfil de orientador, foi deixada uma sugestão que remete novamente para as notificações via *email*. Para este caso, o docente gostaria de ser notificado sempre que uma avaliação relativamente a um trabalho que orienta é submetida.

Finalmente, foi obtida uma única resposta relativamente ao perfil de coordenador. Para este caso, segundo as respostas obtidas, o docente não se deparou com nenhum problema nem durante o processo de criação de épocas de avaliação, nem com o processo de atribuição de júris ou a visualização e publicação de avaliações submetidas pelos avaliadores. Quanto a este perfil, o docente apenas sugeriu a possibilidade de atribuir júris a trabalhos que são entregues tardiamente.

8. Conclusões e trabalhos futuro

Durante o decorrer deste trabalho, o grupo documentou todos os problemas pré-existentes no repositório da plataforma, documentou também um novo requisito para a plataforma tornando-a numa ferramenta mais completa, criou também uma bateria de testes que foi capaz de aumentar a cobertura do código de 45% para 86% (que abrange a implementação deste novo requisito), descobriu e documentou novos problemas e melhorias na plataforma e realizou vários inquéritos aos utilizadores de modo a obter *feedback* que incluiu problemas e melhorias que necessitaram de ser atendidas.

Futuramente, poderão ser documentados novos problemas, melhorias e requisitos que possam ser implementados na plataforma, de modo a torná-la numa ferramenta que abrange todo o *workflow* da UC. Adicionalmente, um número considerável de problemas que foram documentados ficaram por resolver, o que também poderá ser incluído em trabalhos futuros.

Em resumo, o grupo acredita que todas as necessidades para este trabalho foram atendidas, deixando também a possibilidade de dar continuação a futuros trabalhos sobre a plataforma, graças à extensa documentação que está incluída neste relatório e no repositório da plataforma.

No geral, o grupo conseguiu realizar todas as tarefas propostas sem qualquer grande impedimento. Foi revisto o processo de implementação de testes automáticos e foram conhecidas novas ferramentas que podem o permitem alcançar. O grupo desenvolveu também capacidades de análise e espírito crítico relativamente a ferramentas de software.

Bibliografia

- [1] L. Gomes, “*Regulamento do Trabalho Final de Curso*”. 2019
- [2] F. Silva, “*Plataforma DEISI - Gestão de TFCs*”. 2018
- [3] F. Silva, “*Plataforma DEISI - Gestão de TFCs*”. 2018
- [4] A. Dasso & A. Funes, “*Verification, Validation and Testing in Software Engineering*”. 2006
- [5] L. Luo, “*Software Testing Techniques*”. 2001
- [6] *JUnit4, v4.13*
- [7] *Spring MVC, v5.18*
- [8] *TestNG, v7.0.0*
- [9] *JBehave, v4.6*
- [10] *Serenity, v2.0.87*

Anexos

# ID	Título	Classificação	Prioridade
1	Limitar o tamanho dos campos dos formulários	<i>Bug</i>	<i>Low</i>
2	Orientador proposto pelos alunos deve poder validar	<i>Enhancement</i>	<i>Medium</i>
3	Exportar dados em <i>excel</i>	<i>Enhancement</i>	<i>Low</i>
4	Professor que faça login não deve ser automaticamente considerado orientador	<i>Bug</i>	<i>Medium</i>
6	Nas propostas / candidaturas deve ser indicado se algum aluno / grupo não tem inscrição / propôs outros temas	<i>Enhancement</i>	<i>Low</i>
7	Quando se recusa uma proposta o preenchimento de orientador não devia ser obrigatório	<i>Enhancement</i>	<i>Low</i>
8	Quando um trabalho é submetido em grupo não é criada a inscrição	<i>Bug</i>	<i>High</i>
9	Validar números de alunos em grupo	<i>Bug</i>	<i>Medium</i>
11	Validação de propostas de alunos	<i>Bug</i>	<i>Medium</i>
12	Anulação de temas	<i>Bug</i>	<i>Low</i>
13	Filtrar histórico por período de avaliação	<i>Enhancement</i>	<i>Low</i>
14	Ver candidaturas a propostas próprias	<i>Enhancement</i>	<i>Low</i>
15	Atribuição de TFC com múltiplas candidaturas	<i>Enhancement</i>	<i>Medium</i>
16	(Des)Ativação de docentes	<i>Enhancement</i>	<i>Low</i>
17	Listar propostas por status	<i>Enhancement</i>	<i>Low</i>
18	No histórico, os filtros devem ser cumulativos	<i>Enhancement</i>	<i>Low</i>
19	Filtrar candidaturas por candidato	<i>Enhancement</i>	<i>Low</i>
20	Inscrição em grupo com número de aluno inválido não mostra erro	<i>Bug</i>	<i>High</i>
21	Listar candidaturas por perioridade de candidato	<i>Enhancement</i>	<i>Medium</i>
22	Aluno continua a ver inscrições do ano anterior	<i>Bug</i>	<i>Medium</i>
23	Mostrar nomes de alunos (além do número)	<i>Enhancement</i>	<i>Low</i>

24	Data na alteração de estado	<i>Bug</i>	<i>Medium</i>
25	Bloqueio de temas	<i>Enhancement</i>	<i>Low</i>
26	As candidaturas de grupo não estão a ser canceladas após a 1ª atribuição	<i>Bug</i>	<i>High</i>
27	Propostas de professores devem permitir indicar um aluno ou grupo	<i>Enhancement</i>	<i>Medium</i>
28	Rever o fluxo da proposta entre submissão e aprovação	<i>Enhancement</i>	<i>High</i>
29	Vulnerabilidade a ataques XSS	<i>Bug</i>	<i>Medium</i>
30	Associar co-orientador aos trabalhos	<i>Enhancement</i>	<i>Medium</i>
32	Erro ao aceitar uma proposta de TFC feita pelos alunos	<i>Bug</i>	<i>Low</i>
33	Error page não definida	<i>Enhancement</i>	<i>Low</i>
34	Bug ao criar grupo	<i>Bug</i>	<i>Low</i>
36	Adição de tecnologias aceita espaços	<i>Bug</i>	<i>Low</i>
40	Distribuição de TFCs por orientador por disponibilidade do mesmo	<i>Enhancement</i>	<i>Medium</i>

Tabela 3 - Listagem de problemas.

Glossário

Trabalho Final de Curso (TFC) - Unidade curricular dos cursos de Engenharia Informática e Informática de Gestão, onde os alunos devem aplicar os conceitos adquiridos ao longo do curso, num projeto de maior escala.

Departamento de Engenharia Informática e Sistemas de Informação (DEISI) - Departamento da Universidade Lusófona de Humanidades e Tecnologias que pertence à Escola de Comunicação, Arquitetura, Artes e Tecnologias da Informação, responsável pelas Licenciaturas em Engenharia Informática, Informática de Gestão e Engenharia Informática, Redes e Telecomunicações.

ULHT - Universidade Lusófona de Humanidades e Tecnologias.

ECATI - Escola de Comunicação, Arquitetura, Artes e Tecnologias da Informação.

Unidade Curricular (UC) - Unidade de ensino com objetivos de formação próprios e que é objeto de inscrição administrativa e de avaliação traduzida numa classificação final.

Stakeholder - De maneira mais ampla, compreende todos os envolvidos num processo, quer seja de caráter temporário (como um projeto) ou duradouro (como o negócio de uma empresa).

Commit - Em sistemas de controlo de versão, um *commit* adiciona as últimas alterações do código-fonte ao repositório.

Integrated Development Environment (IDE) - Trata-se de uma plataforma que reúne características e ferramentas de apoio ao desenvolvimento de *software*.