

Mestrado em Engenharia Informática
Estágio
Relatório Intermédio

Solução Web para Gestão de Prémios Científicos

João Silas Dias Lopes

jslopes@student.dei.uc.pt

Orientador:

Álvaro Manuel Reis da Rocha

Data: 1 de Julho de 2016



FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Resumo

Atualmente é comum instituições entregarem prémios a pessoas relacionadas com o meio científico. Estes prémios servem como estímulo à inovação científica e ao rigor do trabalho de investigação. São atribuídos a trabalhos de elevada qualidade, sendo avaliados por júris experientes no contexto do prémio.

A participação neste tipo de prémio exige o envio de vários documentos por parte do candidato no envio da candidatura. A quantidade de documentos necessária para uma candidatura torna o processo de análise demorado para o responsável do concurso, principalmente se o candidato quiser modificar a sua candidatura. Para além disso, existe a necessidade da troca de e-mails e ficheiros relacionados com as classificações que o júri atribuiu às candidaturas.

Este estágio teve como principal objetivo o estudo e desenvolvimento de uma aplicação web em PHP com recurso a uma base de dados MySQL. Esta é capaz de lidar não só com os problemas descritos anteriormente, mas também consegue abranger todo o processo da gestão da atribuição de um prémio científico, tendo em conta as diferentes necessidades do administrador, júri e candidato. O administrador da plataforma é o responsável pela organização do prémio.

A aplicação resultante foi testada e satisfaz os requisitos pretendidos pelo cliente, no entanto poderá sofrer modificações no futuro.

Palavras-Chave

Gestão de prémios, Prémio científico, Candidatura, Júri, Dashboard, Aplicação web, PHP, MySQL, Model-View-Controller

Índice

1. Introdução.....	1
1.1. Contextualização.....	1
1.2. Motivação e objetivos.....	3
1.3. Técnicas de análise.....	3
1.4. Ferramentas usadas.....	4
1.5. Planificação.....	5
1.5.1. Primeiro semestre.....	5
1.5.2. Segundo semestre.....	6
1.6. Metodologia.....	8
1.7. Análise de riscos.....	9
1.8. Estrutura do documento.....	10
2. Estado da arte.....	11
2.1. Análise de soluções concorrentes.....	11
2.1.1 Wizehive - Awards Management.....	11
2.1.2 Judgify.....	12
2.1.3 OpenWater Awards.....	12
2.1.4 Smarter Select.....	12
2.1.5 FluidReview.....	13
2.1.6 Ex Ordo.....	13
2.1.7 OpenConf.....	13
2.1.8 EasyChair.....	14
2.1.9 SmartConfoo.....	14
2.1.10 Conference Management System.....	14
2.2. Análise comparativa.....	15
2.2.1 Concorrentes diretos.....	15
2.2.2 Concorrentes indiretos.....	16
3. Análise de requisitos.....	17
3.1. Análise de processos.....	17
3.2. Análise de utilizadores.....	19
3.2.1. Classes.....	19
3.2.2. Caracterização.....	20

3.3.	User stories	21
3.4.	Casos de uso	21
3.5.	Requisitos funcionais.....	22
3.5.1.	Utilizador.....	22
3.5.2.	Administrador	23
3.5.3.	Júri.....	26
3.5.4.	Candidato	26
3.6.	Requisitos não-funcionais.....	27
3.7.	Prototipagem	28
3.8.	Mockups.....	28
4.	Arquitetura	29
4.1.	Descrição geral da arquitetura.....	29
4.2.	Visão tecnológica	30
4.3.	Vistas.....	31
4.3.1.	Vista lógica	31
4.3.2.	Vista de desenvolvimento.....	34
4.3.3.	Vista física.....	36
4.4.	Modelo da base de dados.....	37
4.5.	Segurança	39
5.	Testes	42
5.1	Testes funcionais.....	42
5.2	Testes de usabilidade.....	44
6.	Resultados finais	47
7.	Conclusões e trabalho futuro	50
	Referências	51

Lista de Figuras

Figura 1 – Calendarização inicial das atividades do primeiro semestre	5
Figura 2 – Diagrama de Gantt do planeamento inicial do primeiro semestre	5
Figura 3 – Calendarização final das atividades do primeiro semestre	6
Figura 4 – Diagrama de Gantt do planeamento final do primeiro semestre	6
Figura 5 – Calendarização inicial das atividades do segundo semestre	7
Figura 6 – Diagrama de Gantt do planeamento inicial do segundo semestre.....	7
Figura 7 – Calendarização final das atividades do segundo semestre.....	7
Figura 8 – Diagrama de Gantt do planeamento final do segundo semestre	7
Figura 9 – Diagrama da metodologia em cascata	8
Figura 20 – Diagrama de fluxo dos processos	18
Figura 21 – Diagrama de estados da candidatura	19
Figura 22 – Diagrama da prototipagem.....	28
Figura 23 – Esquema geral do padrão Model-View-Controller	29
Figura 24 – Visão tecnológica.....	30
Figura 25 – Diagrama de atividade do processo de registo e autenticação.....	32
Figura 26 – Diagrama de atividade do processo de convite.....	32
Figura 27 – Diagrama de atividade da adição de prémio.....	33
Figura 28 – Processos gerais entre os atores do sistema	34
Figura 29 – Vista de desenvolvimento	35
Figura 30 – Vista física.....	37
Figura 31 – Modelo da base de dados	38
Figura 32 – Facilidade de aprendizagem do sistema	46
Figura 33 – Critérios do inquérito dos testes de usabilidade.....	46
Figura 34 – Página principal da aplicação	47
Figura 35 – Mensagem de aviso ao utilizador	47
Figura 36 – Adicionar prémio.....	47
Figura 37 – Gestão de utilizadores da plataforma	48
Figura 38 – Candidaturas aprovadas.....	48
Figura 39 – Página com a informação da candidatura	48
Figura 40 – Submissão de candidatura	49

Figura 41 – Painel de avaliação da candidatura 49

Lista de Tabelas

Tabela 1 – Lista de prémios analisados 2

Tabela 2 – Risco #1..... 9

Tabela 3 – Risco #2..... 9

Tabela 4 – Risco #3..... 9

Tabela 5 – Risco #4..... 10

Tabela 6 – Comparação de concorrentes diretos..... 15

Tabela 7 – Comparação de concorrentes indiretos 16

Tabela 8 – Caracterização do administrador 20

Tabela 9 – Caracterização do júri 20

Tabela 10 – Caracterização do candidato..... 20

Tabela 11 – Exemplo de user storie..... 21

Tabela 12 – Exemplo de caso de uso 22

Tabela 13 – Requisitos funcionais do utilizador 23

Tabela 14 – Requisitos funcionais do painel do administrador..... 26

Tabela 15 – Requisitos funcionais do painel do júri..... 26

Tabela 16 – Requisitos funcionais do painel candidato 27

Tabela 17 – Requisitos não-funcionais 28

Tabela 18 – Classes do core da aplicação..... 36

Tabela 19 – Exemplo de um teste funcional 42

Tabela 20 – Testes funcionais do utilizador 43

Tabela 21 – Testes funcionais do painel do administrador..... 44

Tabela 22 – Testes funcionais do painel do júri..... 44

Tabela 23 – Testes funcionais do painel do candidato 44

Tabela 24 – Tarefas dos testes de usabilidade 45

Lista de Anexos

Anexo A – User stories

Anexo B – Casos de uso

Anexo C – Mockups

Anexo D – Testes

Capítulo 1

1. Introdução

Este projeto enquadra-se na disciplina de estágio do Mestrado de Engenharia Informática no ramo de Engenharia de Software e foi realizado sob a orientação do Professor Dr. Álvaro Rocha.

O primeiro capítulo está dividido em oito secções. Inicialmente é apresentado todo o contexto do problema, os fatores mais determinantes para o desenvolvimento do projeto e objetivos gerais, depois as técnicas de análise, as ferramentas usadas e a planificação para o primeiro e segundo semestre. De seguida vem a descrição da metodologia usada e dos riscos associados ao desenvolvimento da solução. Por fim é apresentada a estrutura do documento e o conteúdo de cada capítulo.

1.1. Contextualização

Muitas instituições nacionais e internacionais promovem concursos para premiar trabalhos de tese de mestrado ou doutoramento relacionados com determinado ramo científico. Existem também conferências científicas anuais de muitas áreas de estudo onde são atribuídos prémios científicos a trabalhos publicados e apresentados. Exemplos disso são as conferências ICRA^[1], ICC^[2], e ICASSP^[3], onde são atribuídos variados prémios pela organização responsável, o IEEE.

O prémio “Melhor Tese de Doutoramento Ibérico em STI”^[4] é atribuído anualmente pela Associação Ibérica de Sistemas e Tecnologias de Informação (AISTI) e foi usado como elemento principal no estudo dos processos associados à gestão de prémios científicos. No entanto, para abranger um contexto maior, foram usados outros exemplos. Foi efetuada uma pesquisa que abrangeu prémios nacionais e internacionais do meio científico. Os prémios em que não há a necessidade de troca de ficheiros não foram contemplados na análise. A tabela abaixo mostra todos os regulamentos de prémios que foram analisados.

Nome do Prémio	Organização
Prémio Melhor Tese de Doutoramento Ibérico em STI ^[4]	Associação Ibérica de Sistemas e Tecnologias de Informação
Prémio Melhor Artigo Científico ^[5]	Faculdade de Medicina da Universidade de Coimbra - Gabinete de Apoio à Investigação
Prémio Científico IBM ^[6]	IBM Portugal
Prémio Científico Mário Quartin Graça ^[7]	Casa da América Latina e Banco Santander Totta
Prémio Científico Sesimbra ^[8]	Câmara Municipal de Sesimbra
Prémios Científicos do ISCTE-IUL ^[9]	ISCTE - Instituto Universitário de Lisboa
Pierskalla Best Paper Award ^[10]	Institute for Operations Research and the Management Sciences

CSS Junior Scientific Award ^[11]	Complex Systems Society
Dijkstra Prize ^[12]	European Association for Theoretical Computer Science
Eppendorf Award for Young European Investigators ^[13]	Eppendorf
ICRA Cognitive Robotics Best Paper Award ^[14]	IEEE Robotics & Automation Society

Tabela 1 – Lista de prémios analisados

Depois de analisados os regulamentos, concluiu-se que a maior parte destes prémios são atribuídos anualmente pelas instituições que os suportam. Alguns regulamentos indicam que prémios que são atribuídos em conferências científicas têm características de participação diferentes, pois para participar é necessário fazer parte da instituição que gere o evento, e os trabalhos que vão a concurso são escolhidos pelos membros da comissão científica, não havendo lugar para a submissão da candidatura por parte do candidato. Estes prémios não são propriamente úteis ao estudo do desenvolvimento da plataforma uma vez que esta servirá em grande parte para lidar de forma prática com a transferência de ficheiros das candidaturas.

No entanto, os prémios que não se enquadram em conferências têm características semelhantes entre si. O critério de participação por vezes é mais alargado, no prémio Eppendorf^[13] por exemplo, podem candidatar-se pesquisadores de toda a Europa ligados à área de estudo do prémio.

De uma forma geral, os prémios cujos trabalhos selecionados não são escolhidos pelos membros da comissão possuem todos prazos para entrega dos documentos da candidatura e são enviados através do e-mail. Apesar dos ficheiros pedidos serem diferentes em cada prémio todos exigem o trabalho que vai a concurso. No prémio da Associação Ibérica de Sistemas e Tecnologias de Informação são pedidos ficheiros como a tese de doutoramento em pdf, a síntese do artigo em formato IEEE, o currículo do autor, a lista de publicações do trabalho, e a digitalização do certificado do grau. No prémio atribuído pelo Gabinete de Apoio à Investigação da Faculdade de Medicina da Universidade de Coimbra^[5] pede que o candidato insira as publicações do trabalho que foram feitas em revistas científicas, e nos prémios científicos do ISCTE-IUL^[9] apenas são aceites trabalhos que tenham sido publicados em revistas científicas internacionais indexadas na Web of Knowledge da Thomson Reuters e na Scopus da Elsevier. Esta questão da inserção das publicações foi identificada em dois prémios, sendo que um deles era o prémio da AISTI.

Nos prémios internacionais analisados, apenas existe referência para a inserção das publicações do trabalho que foram feitas em revistas científicas no prémio da Eppendorf.

Depois da análise, foi possível concluir que os prémios possuem todas regras de funcionamento diferentes. De uma forma geral, existem dois tipos de prémios: os de candidatura livre, em que o participante lê o regulamento, verifica se está em condições de participar e envia a sua candidatura, e os prémios em que os participantes são nomeados pelos membros da comissão que organiza o evento e o prémio. Os mais interessantes para o estudo do desenvolvimento da plataforma são os de participação livre, uma vez que são os mais parecidos com o prémio usado como elemento principal no estudo. Apesar das características iguais, podem diferir em quem pode participar no concurso. Possuem ainda critérios diferentes de aceitação de trabalhos, no entanto todos os processos envolvidos em cada prémio de participação livre, de uma maneira geral, seguem o mesmo formato:

- Divulgação do Regulamento (Prazo, Regras de Participação, Critérios de Aceitação)

- Envio das Candidaturas
- Avaliação
- Divulgação e entrega do prémio

Na primeira fase o organizador do concurso divulga o regulamento do prémio. Por norma é feito no website da organização do concurso e em jornais online. O regulamento tem as regras de apresentação das candidaturas, a lista de documentos necessária, critérios de aceitação, constituição do júri e outros esclarecimentos adicionais. Antes do prazo limite de participação, são enviadas as candidaturas dos participantes através do email referido no regulamento. Depois são aprovados ou rejeitados pelo administrador. Segue-se a avaliação dos trabalhos por parte dos membros do júri depois dos documentos chegarem até eles, e por fim é feita a divulgação e entrega do prémio ao vencedor.

Na secção de análise de processos serão expostas mais detalhadamente todas as atividades relativas à gestão de um prémio científico.

1.2. Motivação e objetivos

Atualmente as instituições e responsáveis pela gestão de um prémio científico não usam nenhuma solução que possa ser instalada em servidor próprio baseada no contexto deste problema. Uma vez que o processo pode mudar em cada prémio, as tarefas tornam-se complexas e desgastantes devido à utilização recorrente do e-mail para enviar ficheiros em algumas fases do concurso. Este problema é comum a todos os intervenientes. Com isso surge a necessidade de centralizar o processo de gestão, desenvolvendo-se uma plataforma que possibilite a interação dos vários atores do sistema.

O desenvolvimento desta plataforma irá ajudar os responsáveis pelos prémios, contribuindo para a agilização e organização da totalidade das fases do concurso. O candidato ao prémio terá a oportunidade de fazer toda a parte do processo de participação no mesmo local e o júri poderá fazer a classificação das candidaturas em conformidade com os critérios de avaliação. No fim não terá de retornar as suas avaliações ao responsável pelo prémio.

O projeto foi desenvolvido dando prioridade às necessidades do representante do “Prémio Melhor Tese de Doutoramento Ibérico em STI” da Associação Ibérica de Sistemas e Tecnologias de Informação.

O objetivo foi desenvolver uma plataforma para que cada instituição a possa instalar no seu servidor e que seja capaz de gerir vários prémios científicos em simultâneo, uma vez que uma instituição pode ter vários concursos.

A plataforma é composta essencialmente por três módulos gerais:

- **Painel do administrador** – painel de funcionalidades do responsável pelo prémio relacionadas com a gestão do mesmo
- **Painel do júri** – painel de funcionalidades do júri para a avaliação de candidaturas
- **Painel do candidato** - painel com possibilidade de participação num prémio em aberto

Para concretizar os objetivos, foi importante que a solução apresentasse uma boa resposta a nível de segurança, usabilidade e modificabilidade devido a mudanças futuras que sejam necessárias efetuar.

1.3. Técnicas de análise

Para o desenvolvimento de um novo sistema é necessário conhecer todo o contexto envolvente do problema e os intervenientes que o utilizarão. As necessidades do utilizador

devem ser analisadas e transpostas para os requisitos do sistema. A análise dos requisitos foi feita recorrendo às atividades seguintes como forma de facilitar o processo de criação dos requisitos funcionais e não funcionais do sistema.

Levantamento dos requisitos

Esta etapa exige comunicação entre os stakeholders para se poder determinar quais serão os requisitos do sistema. Para isto recorreu-se às seguintes atividades:

- Reuniões e troca de e-mails com o cliente
- Análise de processos
- Análise de utilizadores

O primeiro passo foi realizar reuniões com o cliente. A troca de e-mails também foi usada para confirmar alguns detalhes pendentes. Destas reuniões retirou-se a informação necessária para fazer uma análise aos processos envolventes na gestão de prémios científicos e deu para definir bem os parâmetros da caracterização dos utilizadores da plataforma.

Análise dos requisitos

Na conclusão da primeira atividade foi criada uma lista com os potenciais requisitos do sistema, e nesta atividade de análise de requisitos pretendeu-se encontrar as lacunas dos requisitos até então definidos. Para atingir o objetivo desta atividade recorreu-se a:

- User stories
- Mockups
- Entrevista com o cliente

As user stories foram usadas para registar a descrição simples de uma funcionalidade a partir da perspectiva do cliente. Através dos critérios de aceitação de cada user storie foi possível encontrar funcionalidades do sistema subentendidas na descrição do cliente. A elaboração de mockups foi útil uma vez que deu aspecto gráfico aos requisitos funcionais, e permitiu fazer uma discussão dos requisitos mais produtiva. A reunião efetuada serviu para a discussão dos requisitos encontrados e encontrar os que estavam em falta.

Registo dos requisitos

Depois de discutidos foi necessário fazer a documentação dos requisitos encontrados. Para este fim foram usados casos de uso.

1.4. Ferramentas usadas

Ao longo do estágio foram usadas as seguintes ferramentas:

- **Balsamiq Mockups** - esta ferramenta foi usada para a elaboração de mockups.
- **Visual Paradigm** - para a elaboração de diagramas.
- **GanttProject** - foi usado para elaborar a calendarização das atividades do planeamento do estágio.
- **Cacoo** – esta plataforma web foi usada para criar diagramas.
- **Netbeans** - IDE para o desenvolvimento do código da plataforma.
- **MySQL Workbench** - ferramenta usada para o desenvolvimento e desenho da base de dados.

1.5. Planificação

A planificação foi a tarefa inicial do estágio e nesta secção pode ser consultada a do primeiro e segundo semestre. Foi usado um diagrama de Gantt para visualização do calendário de atividades.

1.5.1. Primeiro semestre

As figura 1 e 2 apresentam o planeamento inicial das atividades do estágio para o primeiro semestre cujo objetivo principal era fazer um protótipo não funcional daquilo que seria a plataforma.

Nome	Data de início	Data de fim
• Enquadramento no contexto do problema	14-09-2015	18-09-2015
• Estado da arte	21-09-2015	09-10-2015
▼ • Análise de requisitos	12-10-2015	30-10-2015
• Análise de utilizadores	12-10-2015	16-10-2015
• Análise de processos	19-10-2015	23-10-2015
• Criação de user stories	26-10-2015	30-10-2015
• Definição de requisitos	12-10-2015	30-10-2015
• Arquitetura geral da plataforma	02-11-2015	13-11-2015
▼ • Prototipagem - Versão 1	16-11-2015	15-12-2015
• Alterações - Requisitos	16-11-2015	15-12-2015
• Alterações - Mockups	16-11-2015	15-12-2015
▼ • Prototipagem - Versão 2	04-01-2016	08-01-2016
• Alterações - Requisitos	04-01-2016	08-01-2016
• Alterações - Mockups	04-01-2016	08-01-2016
• Escrita do Relatório Intermédio	02-11-2015	20-01-2016

Figura 1 – Calendarização inicial das atividades do primeiro semestre

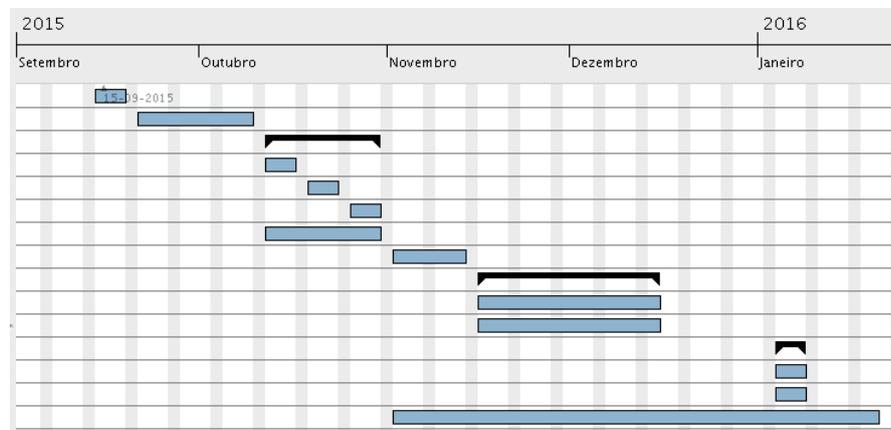


Figura 2 – Diagrama de Gantt do planeamento inicial do primeiro semestre

Após terminado o semestre verificou-se que nem tudo foi cumprido como inicialmente definido. No entanto, a existência de sobreposições nas tarefas é normal uma vez que foram efetuadas em simultâneo. Na figura 3 e 4 é apresentado o planeamento que foi efetivamente executado no decorrer do estágio.

Nome	Data de início	Data de fim
• Enquadramento no contexto do problema	14-09-2015	18-09-2015
• Estado da arte	21-09-2015	09-10-2015
▼ • Análise de requisitos	12-10-2015	13-11-2015
• Análise de utilizadores	12-10-2015	16-10-2015
• Análise de processos	19-10-2015	23-10-2015
• Criação de user stories	26-10-2015	04-11-2015
• Definição de requisitos	12-10-2015	13-11-2015
▼ • Prototipagem – Versão 1	16-11-2015	18-12-2015
• Alterações – Requisitos	16-11-2015	18-12-2015
• Alterações – Mockups	16-11-2015	18-12-2015
▼ • Prototipagem – Versão 2	06-01-2016	15-01-2016
• Alterações – Requisitos	06-01-2016	15-01-2016
• Alterações – Mockups	06-01-2016	15-01-2016
• Arquitetura geral da plataforma	28-12-2015	31-12-2015
• Escrita do Relatório Intermédio	02-11-2015	20-01-2016

Figura 3 – Calendarização final das atividades do primeiro semestre

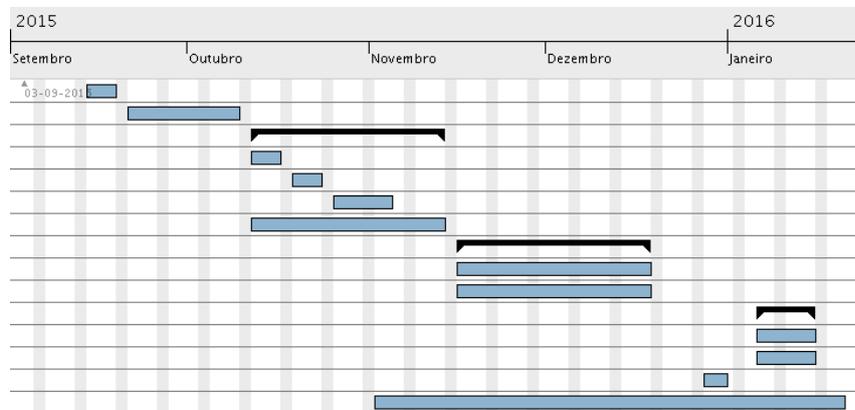


Figura 4 – Diagrama de Gantt do planeamento final do primeiro semestre

Como se pode visualizar na análise de requisitos, a criação de user stories sofreu um atraso porque precisaram de ser modificados após uma reunião com o cliente. A atividade de definição dos requisitos também demorou mais do que o tempo previsto, isto porque existiu mais dificuldade do que o expectável em encontrar alguns requisitos. Com este atraso a atividade de descrição da arquitetura geral foi executada numa semana em que não havia nada agendado. A data da primeira versão do protótipo não funcional sofreu um atraso uma vez que não houve necessidade de ter a primeira versão pronta para entregar ao cliente na semana prevista. Como tal, a segunda versão do protótipo também teve a sua data modificada e o tempo de atividade aumentado, uma vez que o cliente pediu algumas modificações nas funcionalidades.

1.5.2. Segundo semestre

O objetivo do segundo semestre foi desenvolver a aplicação web para a gestão de prémios científicos. No início do segundo semestre o plano que estava feito até então sofreu algumas mudanças. As figuras 5 e 6 apresentam a calendarização inicial das atividades para o segundo semestre do estágio.

Nome	Data de início	Data de fim
Arquitetura	08-02-2016	26-02-2016
• Modelo da base de dados	08-02-2016	19-02-2016
• Segurança	22-02-2016	26-02-2016
Implementação	29-02-2016	13-05-2016
• Estrutura e design	29-02-2016	25-03-2016
• Painel do administrador	28-03-2016	22-04-2016
• Painel do candidato	25-04-2016	06-05-2016
• Painel do júri	09-05-2016	13-05-2016
• Testes	16-05-2016	10-06-2016
• Escrita do Relatório Final	30-05-2016	17-06-2016

Figura 5 – Calendarização inicial das atividades do segundo semestre

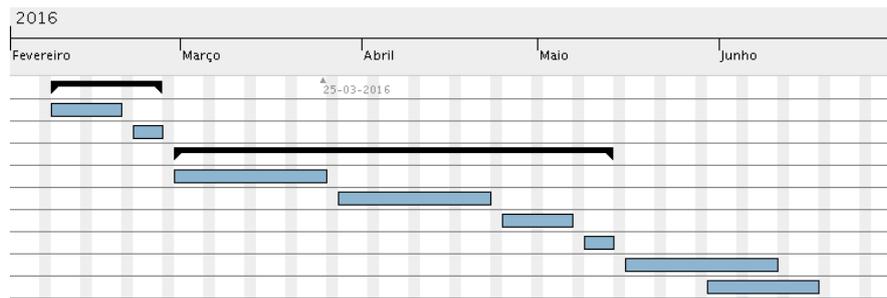


Figura 6 – Diagrama de Gantt do planeamento inicial do segundo semestre

Com o decorrer do estágio verificou-se que algumas tarefas foram mal planeadas. De seguida é apresentado o plano que foi executado durante o estágio.

Nome	Data de início	Data de fim
Arquitetura	08-02-2016	26-02-2016
• Modelo da base de dados	08-02-2016	19-02-2016
• Segurança	22-02-2016	26-02-2016
Implementação	29-02-2016	27-05-2016
• Estrutura e design	29-02-2016	25-03-2016
• Painel do administrador	29-03-2016	21-05-2016
• Painel do candidato	09-05-2016	20-05-2016
• Painel do júri	16-05-2016	27-05-2016
• Testes	30-05-2016	18-06-2016
• Escrita do Relatório Final	13-06-2016	07-07-2016

Figura 7 – Calendarização final das atividades do segundo semestre

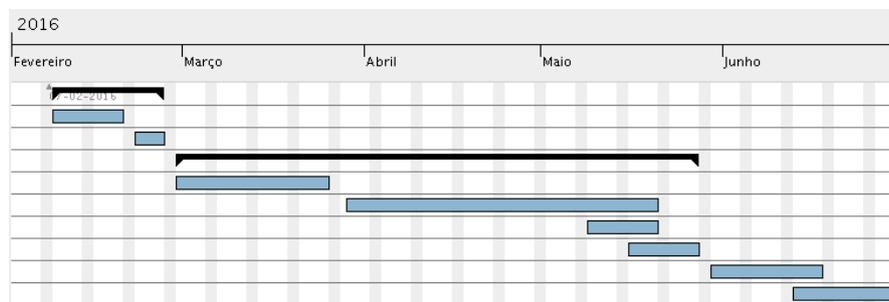


Figura 8 – Diagrama de Gantt do planeamento final do segundo semestre

Como se pode observar na figura 8 a estimativa atribuída à atividade de implementação do painel do administrador foi de quatro semanas, no entanto verificou-se que esse tempo foi insuficiente, tendo sido prolongado por mais duas semanas. Existem tarefas sobrepostas na parte da implementação que não estavam contempladas no plano inicial, isso deveu-se ao peso que têm as precedências dos requisitos durante o

desenvolvimento. Devido ao atraso de duas semanas na implementação não foi possível cumprir as quatro semanas de testes expectáveis. A escrita do relatório final foi terminada durante as três últimas semanas do mês de junho.

1.6. Metodologia

Neste capítulo são descritas as fases da metodologia de desenvolvimento de software usada durante o estágio no desenvolvimento da aplicação. Também são referidas as razões para a escolha do modelo utilizado.

Foi usado o modelo em cascata tradicional com fases bem definidas. Neste modelo a fase seguinte só deve iniciar quando a anterior tiver sido concluída e aprovada pelas partes envolvidas no projeto. Como na parte inicial do projeto existiu um entendimento claro de quais seriam os requisitos do sistema a escolha recaiu num modelo sequencial. Isto facilitou o planeamento das atividades de cada etapa devido à previsibilidade de prazos que este modelo oferece. Na figura 9 está um diagrama que serve de ilustração das fases envolvidas no modelo cascata.

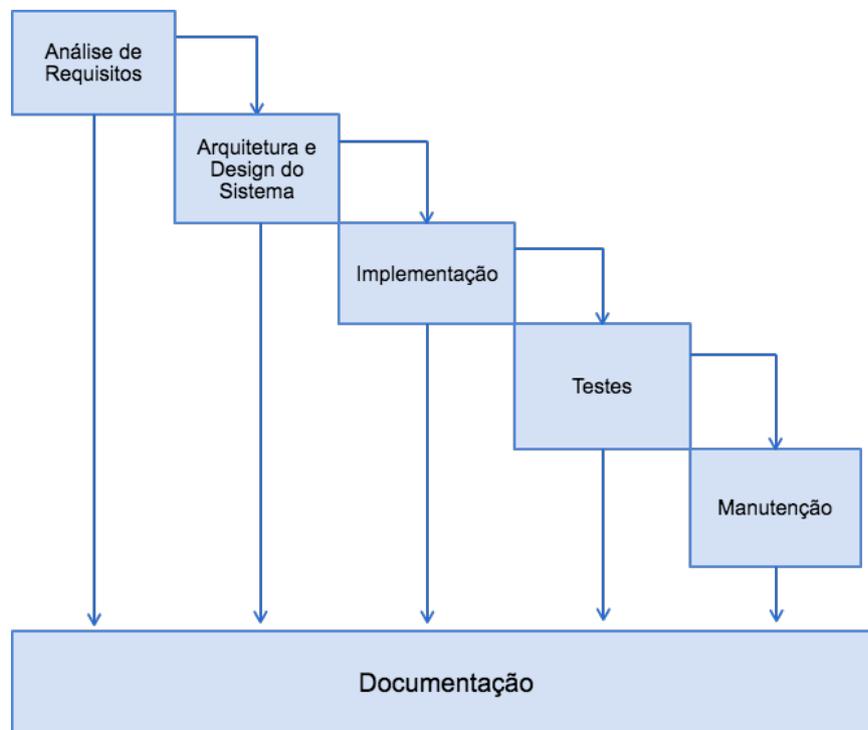


Figura 9 – Diagrama da metodologia em cascata

- **Análise de requisitos** – Nesta fase foi feito todo o trabalho de levantamento, análise e definição dos requisitos do sistema com ajuda dos intervenientes no projeto.
- **Arquitetura e Design do Sistema** – Aqui foi necessário descrever todo o sistema como unidades de software e as suas relações, indicando como o software deve ser implementado e como devia funcionar.
- **Implementação** – Nesta fase foram desenvolvidas as unidades de software.
- **Testes** – Aqui as unidades de software são integradas e testadas.
- **Manutenção** – O sistema foi instalado e colocado em funcionamento. A fase de manutenção pode envolver a correção de erros que possam surgir no ambiente de execução onde vai ser instalado o software.

- **Documentação** – Apesar de não ser uma etapa definida formalmente no modelo em cascata, em todas as fases foi criada documentação com a informação recolhida durante cada fase.

1.7. Análise de riscos

A análise de riscos permite em qualquer projeto de desenvolvimento de software, identificar, avaliar e compreender riscos, que podem condicionar o sucesso do produto final, e desta forma minimizar o impacto dos mesmos pondo em prática o plano de mitigação. Abaixo são mostrados os riscos do projeto em tabelas. Cada tabela contém a descrição do risco, a probabilidade de acontecer (1-5), o impacto (1-5) que poderá vir a ter o risco e por fim o plano de mitigação.

Risco #1 - Erros de planeamento

Descrição	A falta de experiência do estagiário neste tipo de tarefa poderá levar à ocorrência de erros na estimação da duração de determinadas tarefas.
Probabilidade	4
Impacto	3
Plano de mitigação	Fazer uma subdivisão das tarefas maiores em tarefas mais pequenas.

Tabela 2 – Risco #1

Risco #2 - Experiência com a linguagem PHP

Descrição	A pouca experiência do estagiário na utilização da linguagem PHP com o paradigma orientado a objetos pode fazer com que determinadas tarefas demorem mais do que o estimado.
Probabilidade	4
Impacto	2
Plano de mitigação	Estudo de informação científica sobre o desenvolvimento de aplicações web em PHP com recurso ao paradigma de orientação a objetos.

Tabela 3 – Risco #2

Risco #3 – Experiência com jQuery

Descrição	A pouca experiência com a biblioteca javascript jQuery pode fazer com o desenvolvimento da interface do utilizador seja mais demorada.
Probabilidade	4
Impacto	2
Plano de mitigação	Estudo do funcionamento da biblioteca para realizar determinadas operações.

Tabela 4 – Risco #3

Risco #4 – Compreensão do contexto do problema

Descrição	O contexto do problema é relacionado com o meio académico e de pesquisa científica. Para o desenvolvimento da plataforma é necessário conhecer os detalhes de funcionamento da divulgação científica.
Probabilidade	3

Impacto	1
Plano de mitigação	Estudo de informações relativas ao contexto, tais como: funcionamento das publicações em revistas científicas, rankings de revistas e regras de funcionamento dos prémios.

Tabela 5 – Risco #4

1.8. Estrutura do documento

Este documento está dividido nos seguintes capítulos:

- **Introdução** – aqui é exposto todo o contexto envolvente do problema, as motivações e objetivos, técnicas usadas na análise de requisitos, ferramentas usadas, planeamento das atividades, a metodologia e a análise de riscos.
- **Estado da arte** – capítulo dedicado à análise de soluções existentes no mercado concorrentes da plataforma.
- **Análise de requisitos** – este é o capítulo dedicado ao levantamento de requisitos, sendo analisados os utilizadores da plataforma e os processos envolvidos na gestão da atribuição de um prémio. Depois são apresentados os user stories, os requisitos não funcionais e os funcionais com os respectivos casos de uso e mockups desenvolvidos.
- **Arquitetura** – neste capítulo é mostrada a arquitetura geral a seguir na implementação, assim como todos os detalhes necessários para a compreensão do funcionamento da aplicação web.
- **Testes** – este capítulo será dedicado à qualidade do software. Aqui serão mostrados todos os testes feitos à aplicação.
- **Resultados Finais** – aqui são mostrados os resultados mais importantes do desenvolvimento da plataforma.
- **Conclusões** – resumo e análise ao trabalho desenvolvido ao longo do estágio.

Capítulo 2

2. Estado da arte

Este capítulo tem por objetivo a análise de soluções existentes no mercado que possam de alguma forma relacionar-se com o estudo do desenvolvimento da plataforma para a gestão de prêmios científicos.

2.1. Análise de soluções concorrentes

Para o estudo foram contemplados como concorrentes diretos os serviços já existentes que oferecem a gestão de prêmios científicos. Para além destes, foram incluídos softwares e serviços para gestão de conferências que podem ser considerados concorrentes indiretos. Estas soluções são relevantes para o estudo pois possuem partes do processo de participação e interação, semelhantes aos processos da plataforma para a gestão de prêmios. Nas conferências, os candidatos também têm de submeter os ficheiros necessários para a participação. Esses ficheiros devem ser analisados antes da conferência ter início, com objetivo de escolher os mais adequados. Importa salientar que a pesquisa abrangeu as soluções que aparentassem mais maturidade de projeto e documentação.

De seguida serão especificadas as características e funcionalidades mais relevantes de cada solução estudada. Em relação aos concorrentes diretos foram analisados os serviços cloud-based Wizehive Awards Management^[15], Judgify^[16], OpenWater com os módulos Awards Judging^[17] e Online Submissions^[18] e Smarter Select^[19]. Os concorrentes indiretos são o serviço de gestão de processos de candidaturas FluidReview^[20], o Ex Ordo^[21] um serviço de gestão de formulários personalizados que é usado para conferências e submissões de artigos. Por fim foram também analisados softwares e serviços cloud-based idealizados para conferências científicas, como o OpenConf^[22], EasyChair^[23], SmartConfoo^[24] e Conference Management System^[25].

2.1.1 Wizehive - Awards Management



Figura 10 – Logotipo do Wizehive

O WizeHive é um serviço que possui três soluções distintas. Para esta análise foi apenas analisada a documentação da solução de gestão de prêmios. O WizeHive Awards Management é um módulo do software construído para automatizar o processo de recolha, análise e gestão das candidaturas a prêmios, bem como a gestão dos detalhes do prémio a concurso. O sistema funciona permitindo ao administrador construir um formulário personalizado para recolher dados dos participantes. Depois é possível analisar esses dados, e tomar uma decisão em relação aos mesmos. O objetivo do sistema é permitir uma interação com candidatos e júris para assim automatizar todo o processo. Para isto existem tarefas como visualizar, ordenar e organizar submissões, um gestor de tarefas, possibilidade de envio de email para os candidatos individualmente e em grupo, processo de avaliação colaborativo e critérios de avaliação.

O software é cloud-based e por isso não requer nenhuma instalação. Possui uma licença comercial com três tipos diferentes: Standard, Professional, Enterprise. A diferença entre elas é a quantidade de funcionalidades disponíveis e o preço, que não está disponível no website do serviço.

2.1.2 Judgify



Figura 11 – Logotipo do Judgify

O Judgify é um serviço cloud-based preparado para acessos de dispositivos móveis, que permite a organização de prémios, competições e concursos. O administrador pode inserir as regras do prémio, escolher a categoria e os requisitos para as submissões a efetuar pelos candidatos. Disponibiliza método de pagamento, caso seja necessário no evento. Faz a recolha e processamento da informação inerente às candidaturas para depois estas serem avaliadas pelo júri através dos critérios de avaliação. Os resultados das candidaturas são calculados e reportados automaticamente pela plataforma.

Esta plataforma oferece uma licença grátis para eventos de prémios que também sejam grátis, caso contrário cobra 3.5% pelo preço da submissão que o candidato pagar.

2.1.3 OpenWater Awards



Figura 12 – Logotipo do OpenWater Awards

O OpenWater Awards é um serviço de criação de formulários online para os prémios. Possui funcionalidades de submissão de ficheiros de texto ou pdf, gestão de avaliação por parte de júris, comunicação facilitada com os intervenientes, ferramentas administrativas que oferecem visibilidade sobre a totalidade do processo referente ao prémio a decorrer e o utilizador ainda pode usufruir de um menu para a construção do website do prémio. Assim que os candidatos submetem as suas candidaturas podem fazer um pagamento à entidade reguladora do prémio. O serviço possui integração com o Paypal.

A documentação facilitada pelo serviço é muito limitada nos pormenores sobre as características de funcionamento uma vez que se trata de uma ferramenta paga. Contudo, o preço não é divulgado no website da empresa.

2.1.4 Smarter Select



Figura 13 – Logotipo do Smarter Select

O Smarter Select é um serviço cloud-based que permite submissão de candidaturas online de forma simples. É possível recolher os dados previamente escolhidos para o formulário, analisá-los, avaliá-los, criar inquéritos para fazer aos candidatos. O sistema permite a entrada de quatro atores diferentes: “applicant or student”, “recommender”, “evaluator” e o “administrator”. O “recommender” é o responsável por gerar as nomeações para o prémio, o “evaluator” é o júri e o “administrator” é o organizador do prémio. Esta solução possibilita a partilha do link para a

submissão no formulário para aqueles organizadores de prémios que querem divulga-lo nas suas páginas, e recebem as candidaturas no serviço. Este serviço também não possui documentação sobre as características de funcionamento e tem uma versão grátis muito limitada. O preço da versão para 500 utilizadores é de 1000 dólares, para 2500 utilizadores é de 2500 e para utilizadores ilimitados é de 4000 dólares.

2.1.5 FluidReview



Figura 14 – Logotipo do FluidReview

O FluidReview permite personalizar uma página web para o processo de candidatura. Pode configurar todas as regras para a participação e envio de candidaturas, inserir o regulamento do concurso, fazer tracking das visitas com o Google Analytics, login seguro para candidatos e

reviewers através do Twitter, Facebook e LinkedIn e alterar o logótipo associado ao concurso. A plataforma também oferece a possibilidade de atribuir candidaturas para avaliação a determinados reviewers e criar equipas de reviewers. É possível ver os logs com a atividade das candidaturas e os intervenientes no sistema podem falar de forma facilitada. A plataforma facilita o envio massivo de emails para grupos de utilizadores e a criação de um portal para os reviewers poderem fazer a avaliação das candidaturas usando formulários de critérios de avaliação personalizados. Todas as classificações atribuídas são automaticamente ordenadas, o que permite que possam ser vistos os vencedores rapidamente.

2.1.6 Ex Ordo



Figura 15 – Logotipo do Ex Ordo

O Ex Ordo é um software cloud-based que permite a criação de formulários para a submissão de artigos em conferências. Na inserção da candidatura o autor pode colocar diversas informações sobre a candidatura, tais como:

formato, o título, resumo, os autores, os tópicos onde se insere o trabalho e informação adicional. O autor pode editar a sua candidatura até que o prazo para a mesma termine. O serviço permite a exportação de todos os dados relativos a uma conferência para o formato Excel, a criação de um widget para os reviewers avaliarem as submissões enviadas a partir dos critérios e com comentários textuais. Os reviewers podem desempenhar diversos papéis para assim tornar o processo de análise o mais flexível possível. Para além disto o administrador tem acesso às candidaturas avaliadas ordenadas automaticamente pela pontuação obtida na avaliação. A plataforma também permite a criação de formulários para aceitar o pagamento de taxas através de pagamentos por cartão de débito ou Paypal.

Trata-se de um serviço com licença comercial sendo que o preço depende das características escolhidas no momento da criação da conta. A documentação fornecida pelo serviço é escassa e não ajuda a perceber como são as funcionalidades.

2.1.7 OpenConf



Figura 16 – Logotipo do OpenConf

A versão do OpenConf analisada foi a “Community Edition”, que suporta apenas as funcionalidades básicas como submissão de artigo por parte do

candidato, e a revisão e pontuação de artigo por parte do júri. Para além destas funcionalidades conta também com a possibilidade de atribuir artigos a júris de maneira automática ou manual. Possibilita a exportação de dados em diversos formatos, tem um sistema para a recuperação

do login e notificações por e-mail. Esta versão pode ser descarregada gratuitamente e instalada com recurso a um servidor como o Apache. Este software foi desenvolvido em PHP e faz uso de uma base de dados que pode ser MySQL ou MariaDB. Apesar de se tratar de um software para conferências foi o único software analisado que pode ser descarregado e corrido num servidor próprio. Para além da versão analisada existe também o “Professional Edition” que acrescenta mais funcionalidades ao software.

2.1.8 EasyChair



Figura 17 – Logotipo do serviço EasyChair

O EasyChair é um serviço em que o utilizador não tem de fazer instalação num servidor próprio, pois trata-se de uma plataforma que corre nos próprios servidores da EasyChair. Tem disponíveis opções como gestão e monitorização do programa da conferência, possibilidade de troca de mensagens entre diferentes participantes (juízes, autores, membros), discussão online dos trabalhos submetidos, etc. A análise ficou um pouco mais restrita uma vez que não há possibilidade de saber um pouco mais da arquitetura uma vez que para poder usar basta um registo na plataforma. O serviço tem três licenças diferentes: “Free”, “Professional”, e “Executive”. Apesar de se tratar de um serviço muito utilizado no meio científico, não ajuda quem precisa de um serviço associado a atribuição de prémios.

2.1.9 SmartConfoo



Figura 18 – Logotipo do serviço SmartConfoo

resumos e artigos submetidos.

Trata-se de um software que possui uma licença comercial não divulgada no site. A documentação disponível é insuficiente para poder analisar mais detalhes sobre o funcionamento do fluxo do processo de gestão.

2.1.10 Conference Management System



Figura 19 – Logotipo do serviço Conference Management System

O SmartConfoo é um software cloud-based para gestão de conferências académicas. Com este serviço podem ser definidos administradores para a conferência, convidar reviewers para integrarem a equipa que vai avaliar os artigos submetidos e definir templates de e-mail para facilitar a comunicação com os intervenientes.

Todo o fluxo de participação é gerido pela plataforma, indo desde o momento de submissão da candidatura, passando pela revisão até ao estado de aceitação do artigo. Para uma gestão eficaz do evento o software mostra vários dados estatísticos no painel do responsável pelo evento, e ainda oferece um menu para a gestão simplificada de utilizadores, gestão de

O COMS é um software cloud-based para gestão de conferências. A serviço suporta o processo desde o momento do registo, pagamento, submissão de trabalhos e avaliação. Os utilizadores do sistema são

divididos por: “chair”, “administradores”, “reviewers”, “editors” e “authors”. De seguida irão ser referidas as funcionalidades mais importantes de cada ator do sistema.

O administrador tem acesso a todas as funcionalidades de gestão de utilizadores, do sistema, nomeações de assistentes, download dos artigos científicos submetidos, comunicar por meio de templates de e-mail para determinados grupos de utilizadores, criar a data de limite para o registo, envio de artigo e revisão do mesmo. Os editores têm como função principal poderem editar os detalhes do autor e da submissão. Podem também ver a lista de candidaturas ordenada por diversos critérios. Os autores podem submeter, retirar e editar candidaturas, fazer o upload dos ficheiros dos formatos dos artigos pedidos, alterar ficheiros submetidos por outros, ver os resultados das avaliações. Os revisores ou júris podem ver todos os artigos submetidos, estabelecer preferências de avaliação em relação às temáticas dos artigos e ver detalhes dos autores. Cada ator destes tem ainda outras funcionalidades que não são particularmente interessantes para a gestão de um prémio científico.

Esta solução tem uma licença comercial e o seu preço varia consoante as características do serviço que pretender.

2.2. Análise comparativa

A análise comparativa das soluções foi feita recorrendo a tabelas que possuem métricas de comparação. Das métricas escolhidas, fazem parte o tipo de software que é, ou seja, se corre em servidor próprio ou é um serviço cloud-based, o tipo de licença e algumas funcionalidades que foram implementadas na solução desenvolvida.

2.2.1 Concorrentes diretos

Para concorrentes diretos foram avaliadas as soluções Wizehive, Judgify, OpenWater e Smarter Select. Atualmente, estas são as soluções mais interessantes existentes no mercado para a gestão de prémios.

Métrica	Wizehive	Judgify	OpenWater	Smarter Select
Servidor Próprio	X	X	X	X
Licença	Comercial	Grátis*	Comercial	Comercial
Submissão de candidatura	✓	✓	✓	✓
Avaliação da candidatura por um júri	✓	✓	✓	✓
Campos de upload	✓	✓	✓	✓
Formulários de avaliação	✓	✓	✓	✓
Email massivo	✓	✓	✓	✓
Templates para mensagens	✓	✓	✓	✓

Tabela 6 – Comparação de concorrentes diretos

O Judgify serve para gerir eventos e não para gestão de prémios, no entanto caso o evento seja gratuito, o serviço não cobra taxas pela participação de membros, e pode-se adaptar a ferramenta para apenas usá-la para gerir as submissões. A quantidade de funcionalidades adaptada à gestão de eventos pode tornar o processo mais complexo para a pessoa que irá geri-lo. É importante referir que das soluções concorrentes diretas, foi a única que foi possível experimentar, uma vez que todas as outras possuem preços elevados e não dispõem de demonstrações gratuitas. Este género de ferramenta é geralmente adquirida por instituições e empresas que possuam eventos com entrada paga não estando ao alcance de organizações sem fins lucrativos, como por exemplo universidades.

A análise destas ferramentas foi particularmente interessante do ponto de vista de funcionalidades, pois ajudou na clarificação de alguns dos requisitos do sistema desenvolvido. Como pode ser observado ver na tabela, as soluções analisadas possuem as funcionalidades principais desenvolvidas no projeto e consideradas como métricas nesta comparação. No entanto o contexto da funcionalidade pode ser um pouco diferente do contexto real em que foi usada neste projeto.

Em termos arquiteturais não foi possível concluir sobre características da implementação uma vez que a documentação disponibilizada é insuficiente. O Wizehive Awards é o único serviço que disponibiliza documentação sobre as funcionalidades do sistema.

Todos estes serviços são cloud-based, e como já foi referido anteriormente não foi possível identificar soluções que corram em servidor próprio.

2.2.2 Concorrentes indiretos

Como concorrentes indiretos foram considerados os softwares e serviços cloud-based que façam a gestão de conferências. Como já foi referido anteriormente, existem prémios científicos analisados que são atribuídos exclusivamente em conferências. As soluções analisadas foram: FluidReview, Ex Ordo, OpenConf, Conference Management System, EasyChair e o SmartConfoo.

Métrica	Fluid Review	Ex Ordo	Open Conf	COMS	Smart Confoo	Easy Chair
Servidor Próprio	X	X	✓	X	X	X
Licença	Comercial	Comercial	Grátis*	Comercial	Comercial	Comercial
Submissão candidatura	✓	✓	✓	✓	✓	✓
Avaliação da candidatura por um júri	✓	✓	✓	✓	✓	✓
Campos de upload	✓	✓	✓	✓	✓	✓
Formulários de avaliação	✓	✓	✓	✓	✓	✓
Email massivo	✓	✓	✓	✓	✓	✓
Templates para mensagens	✓	✓	✓	✓	✓	✓

Tabela 7 – Comparação de concorrentes indiretos

O OpenConf é a única solução que corre em servidor próprio e possui uma licença que permite testar uma versão de demonstração. Apesar de permitir adicionar vários tipos de documentos a solução é claramente orientada para gestão de conferências e não serve para concursos de prémios científicos.

Os outros serviços analisados disponibilizam a totalidade das funcionalidades desenvolvidas na aplicação mas tornam-se complexos para quem desejar organizar um prémio científico, e no caso de alguns torna-se mesmo impossível. Estas soluções possuem licenças comerciais, o que as torna impraticáveis para muitas instituições.

Podem concluir-se que neste momento não existem soluções no mercado que corram em servidor próprio. As identificadas correm nos servidores das empresas que oferecem os serviços, o que faz com que a ferramenta desenvolvida seja nova para o mercado.

Capítulo 3

3. Análise de requisitos

Este capítulo tem como objetivo apresentar o processo envolvido no levantamento de requisitos necessários para o desenvolvimento da plataforma. Antes de se proceder a esta atividade foram analisados regulamentos de prémios científicos, foram feitas entrevistas ao cliente no sentido de tentar perceber como é que a plataforma podia ajudar nos processos de gestão e também foram analisadas outras plataformas que podiam desempenhar algumas das funcionalidades pretendidas.

3.1. Análise de processos

Nesta secção serão descritos os processos envolvidos num concurso a um prémio científico. Para esta análise foram incluídos os prémios. Estas atividades refletem o comportamento atual dos futuros utilizadores da plataforma: administrador, júri e candidato.

Divulgação do regulamento do concurso

O regulamento do concurso por norma contém informações como:

- Data limite de participação
- Critérios de aceitação das participações
- Informações sobre o prémio
- Informações sobre o júri
- Documentos necessários para participar
- Email para enviar a participação
- Documento de preenchimento para participação

Depois o organizador insere-o no website da instituição, e divulga-o em outros meios de comunicação, como por exemplo jornais online e redes sociais.

Envio dos documentos da candidatura

No prémio usado como elemento principal para a análise de processos o candidato ao ler o regulamento vê quais são os documentos necessários para poder participar, e envia-os para o email que se encontra no regulamento. No entanto em um dos prémios foi possível averiguar a utilização do LimeSurvey^[26] por parte do responsável pela gestão^[27]. Trata-se de uma plataforma orientada para inquéritos mas que consegue fazer parte do processo da gestão de atribuição de um prémio. Não possui funcionalidades relacionadas com avaliação de candidaturas por parte dos júris.

Verificação das candidaturas submetidas

Conforme as candidaturas vão chegando, é necessário verificar se as participações possuem os documentos que foram exigidos pelo regulamento. Este processo é sem dúvida o que mais demora, pois o organizador tem de verificar manualmente todas as candidaturas e os ficheiros inerentes à mesma através do email. O processo fica mais complexo caso o candidato se engane, e tenha a necessidade de efetuar troca de ficheiros da candidatura.

Aprovar candidaturas para a fase final

Após a verificação de validação da participação, o organizador responsável seleciona as candidaturas que passam à fase final e faz os documentos dessas mesmas candidaturas chegarem aos júris do concurso. Após isto divulga a passagem à fase final aos candidatos finalistas. Apesar desta parte do processo ser difícil de analisar através dos regulamentos dos prémios foi dada prioridade ao conhecimento dos processos associados ao prémio usado como elemento principal do estudo.

Avaliação das candidaturas submetidas

O júri após recepção das candidaturas finalistas tem de as avaliar, e para isso usa um formulário com critérios de avaliação que mudam de concurso para concurso. Esta informação é enviada para o administrador através de um formulário. Mais uma vez, nem todos os regulamentos divulgam os critérios de avaliação usados pelos júris.

Atribuição do prémio

O responsável pelo prémio, após recepção das pontuações obtidas nas questões do formulário já tem informação suficiente para decidir qual a candidatura vencedora. A notificação ao vencedor é feita posteriormente através de email ou então de uma cerimonia onde o prémio esteja inserido.

Na figura 20 é apresentado um diagrama de fluxo que contempla todos os processos analisados nesta secção.

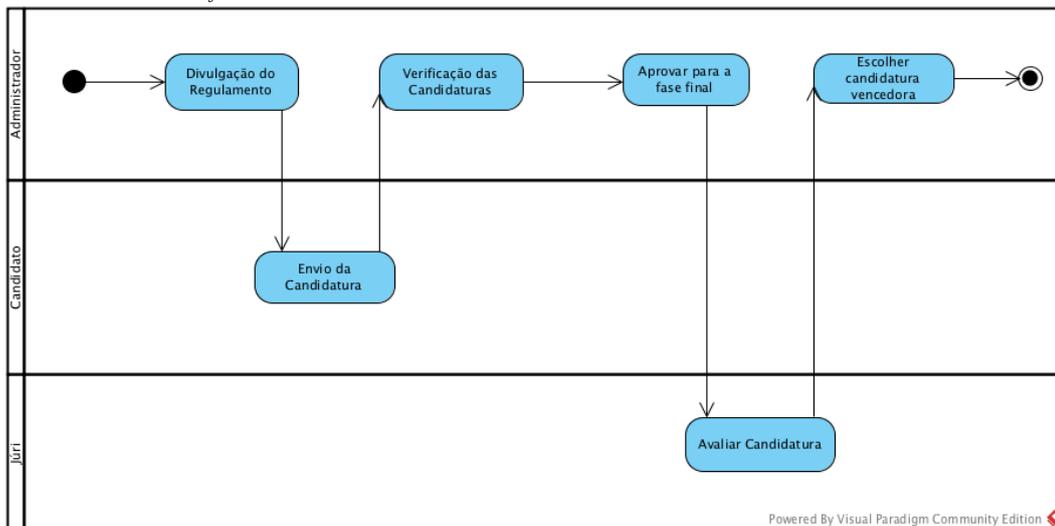


Figura 20 – Diagrama de fluxo dos processos

Na figura 21 está representado um diagrama de estados. Este diagrama demonstra todos os estados que a candidatura pode ter nas várias etapas do fluxo.

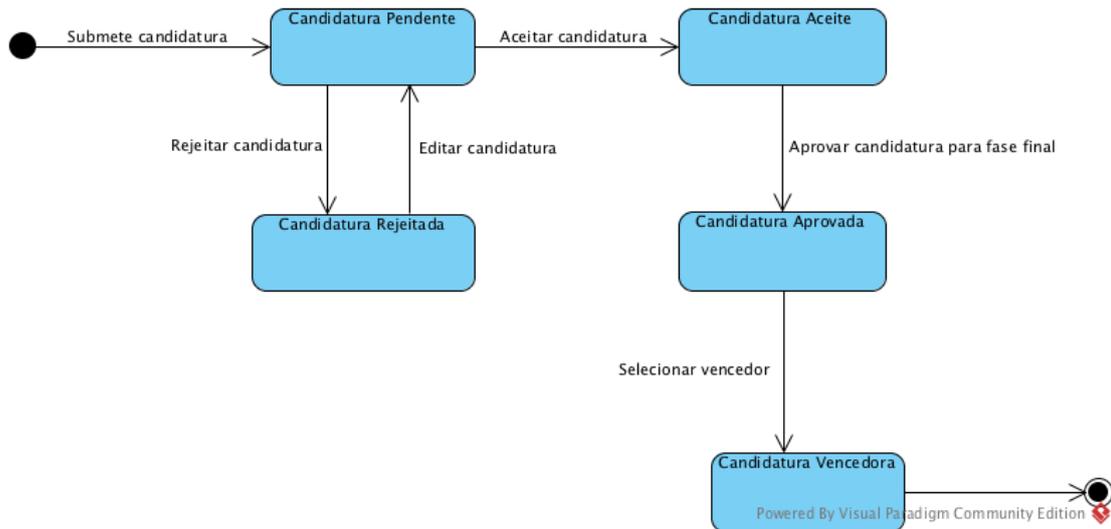


Figura 21 – Diagrama de estados da candidatura

Conclui-se assim que a candidatura a um prémio pode ter cinco estados:

- **Pendente** – trata-se do estado automático atribuído pelo sistema após o envio da candidatura.
- **Rejeitada** – após avaliação o administrador pode rejeitar a participação de determinada candidatura.
- **Aceite** – o administrador poderá aceitar a candidatura para participação se cumprir verificar o cumprimento do regulamento.
- **Aprovada** – após ser aceite a candidatura pode ser aprovada para passar à fase final e só a partir daí pode ser avaliada pelos júris escolhidos para o prémio.
- **Vencedora** – após ser aprovada para a fase final a candidatura pode ser marcada como vencedora.

A participação neste tipo de prémio exige o envio de vários documentos pelo candidato, como por exemplo uma versão completa e resumo do trabalho, currículo vitae, etc. A quantidade de documentos necessária para uma candidatura torna o processo de análise demorado para o administrador do concurso. Por isso mesmo é necessário garantir que com uma plataforma adequada seja possível lidar de forma prática com este problema. A plataforma deve abranger todo o processo desde a introdução do regulamento e recepção dos documentos dos candidatos, à avaliação dos mesmos pelo júri e divulgação do vencedor. A vista lógica da arquitetura no capítulo 4 ajuda à compreensão da maneira como se lidou com as necessidades anteriormente descritas.

3.2. Análise de utilizadores

A plataforma será usada por diferentes tipos de utilizadores. A análise dos utilizadores que irão usar a plataforma é um aspecto importante no desenvolvimento da mesma, pois ajuda na compreensão das necessidades especiais de cada tipo de utilizador. De seguida será feita uma caracterização do perfil de cada tipo de utilizador, olhando para características como: educação, experiência com utilização de computadores e internet, frequência de utilização, dispositivo que utiliza, contexto social e motivação para o uso da plataforma.

3.2.1. Classes

Os utilizadores da plataforma podem ser divididos nas seguintes classes específicas:

- Administrador
- Júri
- Candidato

3.2.2. Caracterização

Identificadas as classes de utilizadores, fez-se a sua caracterização.

Administrador

Educação	Licenciatura, Mestrado ou Doutoramento
Experiência	Média-Alta
Frequência de utilização da plataforma	Alta
Tipo de dispositivo	Tablet, Smartphone, Computador
Contexto social	Membro da organização responsável pelo prémio.
Motivação	Gerir todo o processo do concurso ao prémio de maneira centralizada.

Tabela 8 – Caracterização do administrador

Júri

Educação	Mestrado ou Doutoramento
Experiência	Média-Alta
Frequência de utilização da plataforma	Média
Tipo de dispositivo	Tablet, Smartphone, Computador
Contexto social	Professor, investigador ou um membro da organização do concurso com experiência no ramo a que o prémio pertence.
Motivação	Avaliar os documentos das candidaturas com base nos critérios de avaliação definidos de forma simples sem ter de recorrer ao email.

Tabela 9 – Caracterização do júri

Candidato

Educação	Licenciatura, Mestrado ou Doutoramento
Experiência	Média-Alta
Frequência de utilização da plataforma	Baixa
Tipo de dispositivo	Computador
Contexto social	Estudante ou Professor
Motivação	Participar no concurso a um prémio científico.

Tabela 10 – Caracterização do candidato

Da caracterização feita é importante destacar que o acesso à plataforma também poderá ser feito através de smartphone ou tablet, e por isso o front-end terá de se adaptar a vários dispositivos móveis. A experiência dos utilizadores da plataforma com tecnologias é alta uma vez que o seu contexto social é feito em torno do ramo científico. O responsável pela organização do prémio toma o lugar de administrador no estudo do desenvolvimento da plataforma, e como é lógico será o utilizador que mais utilizará as funcionalidades. O júri como só tem de avaliar as candidaturas que o administrador lhe fizer chegar, a frequência de utilização será média, dependendo do número de candidatos ao prémio em questão. O

candidato apenas irá usar a plataforma para enviar a sua candidatura e por isso a frequência de utilização é baixa.

3.3. User stories

Depois de ouvidas as necessidades do cliente, foram feitas user stories. São descrições simples de uma característica, relatada a partir da perspetiva da pessoa que deseja o recurso. As user stories desenvolvidas seguem o modelo seguinte para a descrição:

Como um <ator>, quero <alguma meta> para que <alguma razão>.

Os critérios de aceitação são usados para especificar como determinada necessidade é ultrapassada pelo sistema. Abaixo está um exemplo de user storie. A lista com todas as user stories criadas pode ser consultada no **Anexo A – User Stories**.

Descrição	Como um utilizador, quero fazer o registo para poder usar a plataforma.
Critérios de Aceitação	<ul style="list-style-type: none"> • O sistema deve permitir fazer o registo como candidato ou júri. • O registo deve pedir o nome do utilizador, e-mail e uma senha.

Tabela 11 – Exemplo de user storie

3.4. Casos de uso

Os casos de uso foram usados para captar detalhes da interação do utilizador com o sistema. Para cada caso de uso foi elaborada uma tabela com os seguintes parâmetros para descrição:

- **ID** – usado para determinar um valor numérico para o caso de uso
- **Ator** – tipo de utilizador (administrador/júri/candidato) envolvido no caso de uso
- **Prioridade** – valor atribuído pelo cliente a determinado caso de uso. Para este efeito usou-se o método MoSCoW^[28]. Tipicamente as categorias de classificação são quatro:
 - **Must** – é a classificação atribuída a um requisito que tem de estar incluído na versão final do produto. Sem ele a solução é considerada um fracasso.
 - **Should** – são considerados os requisitos que devem ser implementados se for possível.
 - **Could** – são os requisitos que são desejáveis pelo cliente, no entanto não são necessários.
 - **Won't** – requisito que não seja incluído na versão final mas que pode ser implementado no futuro.
- **Descrição** – descrição geral do recurso pretendido.
- **Pre-condições** – casos de uso necessários para a execução.
- **Fluxo esperado** – descrição dos passos necessários para a execução do caso de uso.
- **Fluxo alternativo** – descrição das alternativas que podem suceder caso não ocorra o esperado no caso.
- **Pós-condições** – requisitos satisfeitos com a execução do caso.

A lista com todos os casos de uso criados pode ser vista no **Anexo B – Casos de uso**. Abaixo é apresentado o caso de uso de registo na plataforma como exemplo.

ID	#1.1
Ator	Utilizador
Prioridade	Must
Descrição	A plataforma deve ter um formulário de registo. O visitante pode registar-se como candidato ou júri.
Pré-condições	Nenhuma.
Fluxo Esperado	O utilizador insere o seu nome, e-mail e senha e clica no botão de registo.
Fluxo Alternativo	Caso o e-mail já esteja em utilização na plataforma é emitida uma mensagem e o sistema impede a sua utilização. Caso o utilizador não preencha todos os campos é emitido um aviso.
Pós-condições	O utilizador encontra-se com o registo feito na plataforma.

Tabela 12 – Exemplo de caso de uso

3.5. Requisitos funcionais

Nesta secção encontram-se os requisitos funcionais da plataforma representados através de uma tabela com o ID, nome, prioridade e descrição de cada um. Os requisitos identificados foram agrupados pelo ator que o desempenhará.

3.5.1. Utilizador

A categoria utilizador incorpora os requisitos que serão comuns a todos os atores do sistema, excluindo o primeiro requisito uma vez que o administrador não precisa efetuar registo para utilizar a plataforma.

ID	Nome	Prioridade	Descrição
1.1.1	Registar	Must	A plataforma deve ter um formulário de registo público para o candidato.
1.2.1	Login	Must	Para aceder às funcionalidades da plataforma todos os utilizadores têm de fazer a sua autenticação. Ao entrar nas páginas da plataforma é redirecionado para a página de login caso não o tenha feito.
1.3.1	Recuperar senha	Must	A plataforma deverá garantir que os utilizadores recuperem a sua senha caso se não se lembrem qual é.
1.4.1	Editar perfil	Must	O utilizador deve poder editar os dados pessoais. (Nome, senha, email)
1.5.1	Logout	Must	Quando o utilizador tiver terminado as tarefas na plataforma tem de ter maneira de encerrar a sua sessão.
1.6.1	Sessão expirar	Should	Se o utilizador ficar inativo por um período de 20 minutos a sessão deve expirar automaticamente.

1.7.1	Finalizar registo	Must	Qualquer utilizador que seja convidado deve poder finalizar o seu registo através de um link enviado por email, de uma página para o efeito.
1.8.1	Confirmar registo	Must	Quando um candidato se regista deve poder confirmar o seu registo através de um link de confirmação enviado por email.
1.9.1	Ver últimos prémios	Should	A plataforma deve mostrar na sua página inicial algumas informações dos últimos prémios que tenham sido adicionados.

Tabela 13 – Requisitos funcionais do utilizador

3.5.2. Administrador

Nesta secção estão compreendidas todas as funcionalidades atribuídas ao administrador. Este deve ter o controlo sobre a gestão das opções gerais da plataforma, gestão de utilizadores, envio de mensagens, criação de formulários com critérios de avaliação das candidaturas, escolha dos documentos a pedir aos candidatos para participação, etc. Com estas funcionalidades o administrador fica capaz de gerir a atribuição de um prémio científico.

ID	Nome	Prioridade	Descrição
2.1.1	Editar configurações gerais da plataforma	Should	O administrador deve poder configurar algumas opções do funcionamento da plataforma.
2.2.1	Convidar membro para registo	Must	O administrador deve poder convidar júris ou outros administradores para estes efetuarem o seu registo.
2.3.1	Mostrar lista de utilizadores registados	Must	O administrador deve ter acesso a uma lista com todos os utilizadores da plataforma.
2.3.2	Editar utilizador	Must	O administrador deve poder editar os dados de um determinado utilizador da lista.
2.3.3	Remover utilizador	Must	O administrador deve poder remover um determinado utilizador da lista.
2.4.1	Mostrar lista de templates de mensagem	Must	O administrador deve poder ver uma lista com todos os templates de mensagem adicionados por si.
2.4.2	Adicionar template de mensagem	Must	O administrador deve poder adicionar um template de

			mensagem.
2.4.3	Remover template de mensagem	Must	O administrador deve ter a possibilidade de remover um template de mensagem adicionado por si anteriormente.
2.4.4	Editar template de mensagem	Must	O administrador deve poder editar um template de mensagem.
2.4.5	Enviar mensagem	Must	O administrador deve poder enviar mensagens para vários utilizadores com recurso à utilização dos templates de mensagem.
2.5.1	Mostrar lista de formulários de critérios de avaliação	Must	O administrador deve ter acesso a uma lista com todos os formulários criados.
2.5.2	Adicionar formulário de critérios de avaliação	Must	O administrador tem de ter a capacidade de criar formulários de critérios de avaliação.
2.5.3	Remover formulário de critérios de avaliação	Must	O administrador pode remover formulários de avaliação.
2.5.4	Editar formulário de critérios de avaliação	Must	O administrador deve poder editar os formulários de critérios de avaliação.
2.6.1	Mostrar lista de critérios de avaliação	Must	O administrador deve possuir uma lista com todos os critérios de avaliação que foram inseridos na plataforma.
2.6.2	Adicionar critério de avaliação	Must	O administrador deve ter uma opção para adicionar um critério de avaliação.
2.6.3	Remover critério de avaliação	Must	O administrador deve conseguir remover um determinado critério de avaliação adicionado anteriormente.
2.6.4	Editar critério de avaliação	Must	O administrador deve poder editar um critério de avaliação.
2.7.1	Mostrar lista de campos de upload	Must	O administrador deve ter acesso a uma lista com todos os campos de upload adicionados na plataforma.
2.7.2	Adicionar campo de upload	Must	O administrador deve poder adicionar campos de upload para o candidato preencher.
2.7.3	Remover campo de	Must	O administrador deve poder apagar

	upload		um campo de upload que tenha sido adicionado.
2.7.4	Editar campo de upload	Must	O administrador deve poder editar as configurações do campo de upload adicionado.
2.8.1	Mostrar lista de prémios	Must	O administrador deve poder visualizar uma lista completa com os prémios que foram geridos pela plataforma.
2.8.2	Adicionar prémio	Must	O administrador deve poder adicionar um novo prémio preenchendo os campos pedidos.
2.8.3	Remover prémio	Must	O administrador deve poder remover um determinado prémio da lista de prémios.
2.8.4	Editar prémio	Must	O administrador deve ter a possibilidade de editar as configurações e dados relativos ao prémio que inseriu no momento da edição.
2.9.1	Moderar candidaturas pendentes	Must	O administrador deve fazer uma moderação da candidatura enviada para validar a sua participação.
2.9.2	Enviar candidatura para avaliação	Must	O administrador deve ter a capacidade de escolher quais as candidaturas que serão aprovadas para a fase final e portanto para avaliação por parte dos júris.
2.9.3	Ordenar candidaturas avaliadas	Must	O administrador deve ter acesso à visualização de uma lista com todas as candidaturas que tenham sido avaliadas pelo júri em determinado prémio. O ordenamento deve ser pela classificação obtida do júri e pelo rácio das suas publicações.
2.9.4	Remover candidatura	Must	O administrador deve poder remover uma candidatura que esteja no estado pendente.

2.9.5	Escolher candidatura vencedora	Must	O administrador deve poder selecionar a candidatura vencedora na lista de candidaturas avaliadas.
-------	--------------------------------	------	---

Tabela 14 – Requisitos funcionais do painel do administrador

3.5.3. Júri

As funcionalidades do júri são relacionadas com a classificação das candidaturas que foram aprovadas pelo administrador para a fase final.

ID	Nome	Prioridade	Descrição
3.1.1	Mostrar histórico de prémios	Should	O júri deve ter a possibilidade de ver uma lista com os prémios onde já participou como júri.
3.2.1	Mostrar candidaturas para avaliação	Must	O júri deve poder ver uma lista de candidaturas aprovadas para a fase final.
3.3.1	Mostrar lista de avaliações	Must	O júri deve ter acesso a uma lista de avaliações que já fez.
3.4.1	Avaliar candidatura	Must	O júri deve poder classificar as candidaturas com base nos critérios definidos no formulário escolhido para o prémio.
3.4.2	Remover avaliação da candidatura	Must	O júri pode remover avaliação de candidatura enquanto o prazo para avaliação estiver em aberto.
3.4.3	Editar avaliação de candidatura	Must	Enquanto não for encontrado o vencedor o júri pode alterar a sua classificação à candidatura editando o valor dos atribuído aos critérios.
3.4.4	Mostrar ficheiros da candidatura	Must	O júri deve visualizar no browser os ficheiros submetidos pelo candidato no momento em que este fez a sua candidatura.

Tabela 15 – Requisitos funcionais do painel do júri

3.5.4. Candidato

As funcionalidades do candidato refletem as necessidades do processo de candidatura a um prémio científico que ainda esteja em aberto.

ID	Nome	Prioridade	Descrição
4.1.1	Mostrar candidaturas	Should	O candidato deve ter uma lista de candidaturas aos prémios a que concorreu.
4.2.1	Submeter candidatura	Must	O candidato deve poder submeter a sua candidatura a um prémio que esteja em aberto.
4.2.2	Mostrar estado da candidatura	Should	O candidato deve poder visualizar em que estado se encontra a sua candidatura.
4.2.3	Remover candidatura	Must	O candidato deve ter a possibilidade de remover a sua candidatura enquanto esta não tiver sido aceite ou aprovada à fase final.
4.2.4	Editar candidatura	Must	O candidato deve ter a possibilidade de editar a sua candidatura enquanto esta não tiver sido aceite ou aprovada à fase final.

Tabela 16 – Requisitos funcionais do painel candidato

3.6. Requisitos não-funcionais

Os requisitos não-funcionais definem as qualidades gerais e atributos de qualidade do sistema implementado. Abaixo é apresentada a lista com todos os requisitos não-funcionais acordados entre as partes do projeto. Importa salientar que alguns dos requisitos listados tiveram influência direta na escolha arquitetural para a solução. No capítulo 4 é explicado como se lidou com eles no desenvolvimento da solução.

ID	Nome	Descrição
5.1.1	Idioma	O idioma usado na linguagem da plataforma deve ser o inglês.
5.2.1	Segurança	A plataforma deve estar protegida contra SQL injection, bots de spam, vulnerabilidades conhecidas como cross-site request forgery, ou cross-site scripting e ainda usar o protocolo HTTPS.
5.3.1	Modificabilidade	O código desenvolvido deve mostrar características para assegurar a facilidade de alteração ou adição de funcionalidades no futuro.
5.4.1	Usabilidade	É importante que o front-end apresente capacidade de adaptação ao dispositivo usado

na navegação. Deve ser facilitada a possibilidade de desenvolvimento de uma interface gráfica para cada tipo de utilizador.

Tabela 17 – Requisitos não-funcionais

3.7. Prototipagem

A prototipagem é uma técnica usada no desenvolvimento de aplicações que permite que os desenvolvedores entendam de forma clara aquilo que o cliente deseja. O protótipo faz a identificação das funcionalidades e do fluxo da aplicação de forma visual. Através da visualização de como será o sistema, o cliente tem a possibilidade de indicar as modificações a fazer antes da implementação. Trata-se de uma técnica capaz de reduzir os custos do desenvolvimento.

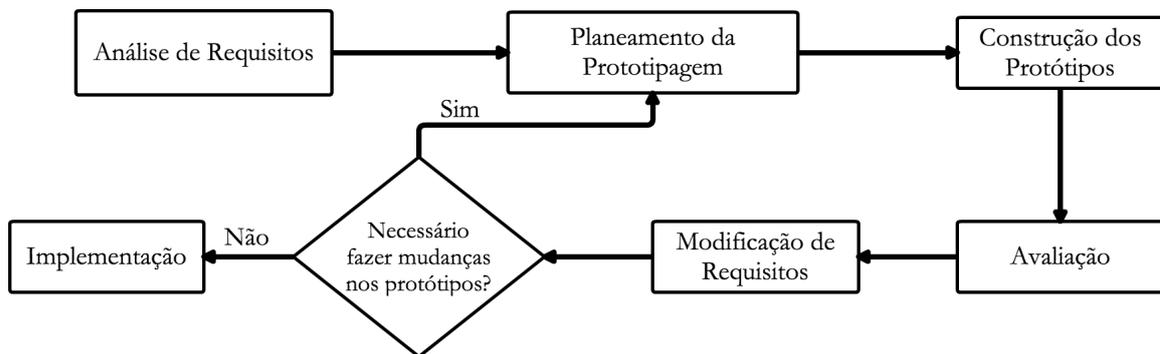


Figura 22 – Diagrama da prototipagem

O fluxo começa com a análise de requisitos do sistema, seguidamente foi feito um planeamento da prototipagem onde foram analisadas separadamente as funcionalidades a desenvolver para cada tipo de utilizador. Depois de feita a construção dos protótipos seguiu-se a avaliação dos mesmos por parte do cliente. Após esta etapa procedeu-se à modificação dos requisitos que estiveram envolvidos nas mudanças pedidas pelo cliente.

3.8. Mockups

A elaboração de mockups foi importante no projeto uma vez que facilitou a discussão dos requisitos nas reuniões com o cliente. Foi também através dos mockups que se conseguiu chegar às versões de protótipos não-funcionais calendarizados para o primeiro semestre do estágio. Todos os mockups elaborados para o projeto podem ser consultados no **Anexo C – Mockups**.

Capítulo 4

4. Arquitetura

Este capítulo tem como principal função descrever a arquitetura da solução implementada. Começa com uma descrição geral de aspectos como os componentes do padrão, as vantagens alcançadas ao utilizá-lo e os motivos principais para desenvolver a aplicação de raiz. Depois é apresentada a visão tecnológica, onde são expostas as tecnologias usadas, as vistas que foram usadas para a documentação da arquitetura, e por fim é apresentado o modelo da base de dados e os aspectos de segurança.

4.1. Descrição geral da arquitetura

Antes de se prosseguir para a descrição da arquitetura, é importante salientar que esta deve cumprir certos atributos de qualidade. Os mais relevantes no estudo da arquitetura foram três: usabilidade, modificabilidade e segurança. A escolha arquitetural podia causar impacto relevante na concretização dos dois primeiros atributos, e por isso optou-se por um padrão que oferece essas qualidades^[29].

O padrão arquitetural escolhido para o desenvolvimento da solução foi o Model-View-Controller. Este padrão é conhecido por simplificar o desenvolvimento e a manutenção da aplicação, uma vez que a divide em três componentes lógicos:

- **Modelo:** O modelo é responsável pela lógica de negócio da aplicação. Faz o encapsulamento do acesso aos dados armazenados na base de dados e fornece uma biblioteca de classes reutilizáveis em outras partes do código da aplicação.
- **Controlador:** O controlador é responsável pela recolha de dados de input vindos do utilizador e decide qual a parte da lógica do negócio que irá ser chamada para execução. Assume a gestão do controlo de fluxo da execução da aplicação. E ainda seleciona a vista que irá ser carregada para mostrar a informação retornada do modelo.
- **Vista:** A vista é a parte encarregue de gerar as páginas web que ficam disponíveis para o utilizador. Recebe a informação do modelo através do controlador e formata-a para a impressão adequada.

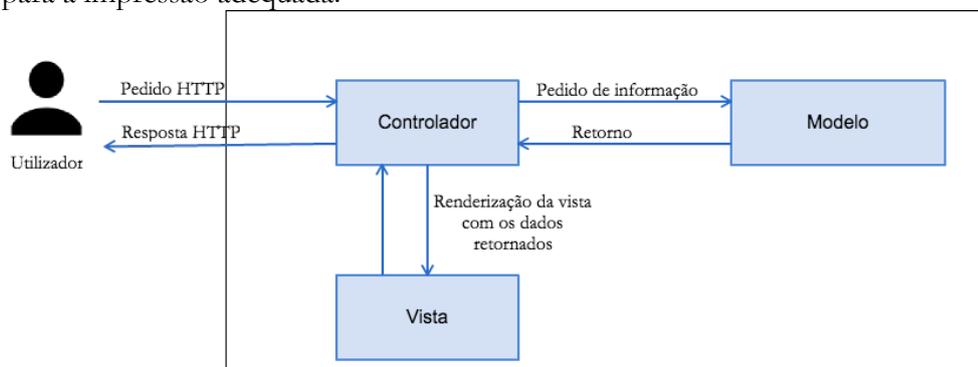


Figura 23 – Esquema geral do padrão Model-View-Controller

Com a sua utilização, é alcançado o fraco acoplamento das partes constituintes da aplicação, que traz vantagens para a modificabilidade do código, uma vez que se for necessário modificar ou adicionar funcionalidades na parte da lógica de negócio não é necessariamente obrigatório modificar a interface. A usabilidade que o MVC garante é a possibilidade de satisfazer necessidades particulares de grupos de utilizadores do sistema com facilidade. Isto

faz com que este padrão seja uma prática comum^[30] no desenvolvimento de aplicações web de interação do utilizador com o sistema.

Atualmente existem muitas frameworks feitas em PHP e usadas para o desenvolvimento de aplicações web, como o Zend Framework^[31], CodeIgniter^[32], CakePHP^[33] e Laravel^[34]. Apesar do padrão MVC ser amplamente usado por este tipo de frameworks, elas apresentam algumas desvantagens que fizeram com que não fosse utilizada nenhuma framework para o desenvolvimento. As desvantagens da utilização destas frameworks passam pela necessidade de atualizações recorrentes, que pode originar problemas de segurança. O código é público e o desenvolvimento da aplicação tem de ser de acordo com a framework, que pode originar problemas futuros de modificabilidade. Como as frameworks CakePHP e Laravel podem trazer problemas^[35] de performance, e o CodeIgniter introduz dificuldade na tarefa de manutenção de código, optou-se por fazer o desenvolvimento de raiz, sem recorrer a nenhuma framework PHP.

4.2. Visão tecnológica

De seguida apresenta-se uma visão que enquadra o produto no seu ambiente tecnológico. A escolha das tecnologias PHP e MySQL para o desenvolvimento da aplicação foram restrições impostas pelo cliente.

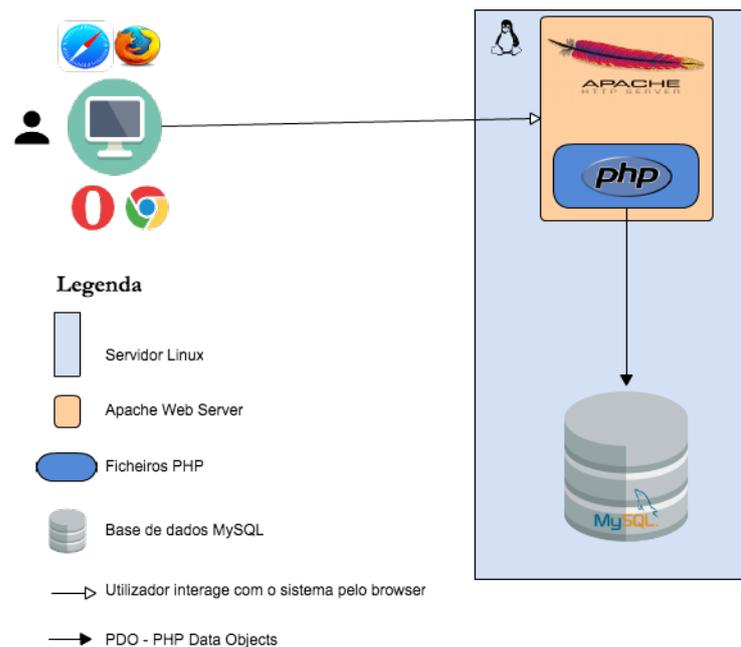


Figura 24 – Visão tecnológica

Como se pode ver na figura 28, os ficheiros constituintes da aplicação irão ser executados pelo Apache HTTP Server com recurso ao módulo PHP e uma base de dados MySQL. O utilizador poderá aceder à aplicação através de um browser num computador. De seguida é apresentada uma lista com todas as tecnologias usadas para o desenvolvimento da aplicação.

- **PHP 5.6.21** - É uma linguagem de programação que oferece suporte à utilização de objetos, corre no lado do servidor e foi projetada para o desenvolvimento de aplicações web. O código PHP pode ser colocado juntamente com HTML e o interpretador é implementado como o módulo do Apache.
- **Apache HTTP Server 2.4.18** - Servidor web que gere os pedidos HTTP vindos do browser dos utilizadores da plataforma.

- **MySQL 10.1.13-MariaDB** - Motor de base de dados relacional responsável pelo armazenamento da informação necessária.
- **HTML 5** – Linguagem interpretada pelo browser para construção de páginas web.
- **CSS** - Usado para formatar o estilo dos componentes da página web.
- **Javascript** – Linguagem interpretada por browsers que permite a criação de scripts do lado do cliente, para que estes controlem o browser e o conteúdo que está a ser apresentado.
- **jQuery 1.12.0** – Biblioteca do Javascript capaz de tornar mais simples a manipulação de código HTML no lado do cliente. É bastante comum a sua utilização uma vez que é suportada pela maioria dos browsers mais usados.
- **Bootstrap 3.3.6** – É uma framework para facilitar o desenvolvimento da interface de aplicações web. É grátis, open-source e contém código HTML de modelos de design já com o CSS definido para a tipografia, formas, botões, navegação e outros componentes da interface. Todo o front-end da aplicação foi construído recorrendo aos componentes^[36] do Bootstrap. A escolha recaiu nesta framework pois é simples de desenvolver a partir dela, possui muitos templates^[37] para estilos diferentes de aplicação ou website e não possui qualquer custo associado devido à licença MIT. Outra vantagem da utilização desta framework é que ela torna o design totalmente adaptado para dispositivos móveis sem grandes custos associados.
- **Swift Mailer 5.4.2** – Esta biblioteca aproxima o programador das funções nativas de envio de emails do sistema onde corre a aplicação. Oferece vários^[38] tipos de envio, através de SMTP, Sendmail ou da função nativa do PHP. A aplicação usa o SMTP uma vez que foi necessário testar as funcionalidades que usavam o envio de emails, e a máquina onde foi desenvolvida não possuía o Sendmail. No entanto o código também possui a função para enviar através do Sendmail, para poder ser usada em qualquer servidor Linux que possua esse agente de transporte de e-mail.

4.3. Vistas

A arquitetura de um software é uma entidade complexa difícil de descrever apenas com uma perspectiva. Uma das práticas mais utilizadas para a documentação de arquiteturas são as vistas^[39]. É comum utilizarem-se várias para descrever a arquitetura de um software. Para a arquitetura da aplicação desenvolvida foram usadas as vistas propostas no modelo “4+1”^[40] de Philippe Kruchten.

4.3.1. Vista lógica

A vista lógica apresenta os comportamentos ou serviços que o sistema deve fornecer aos utilizadores finais. Esta vista não se prende a questões de desenvolvimento e implementação, mostrando apenas a lógica de funcionamento do sistema e como ele processa a informação introduzida. Nas figuras que se seguem são apresentados diagramas de atividade com a lógica de funcionamento da aplicação. Começa por ser apresentado o processo de registo e autenticação, depois a criação do prémio e por fim a participação do júri e candidato no prémio criado. Desta forma é possível apresentar de uma forma abrangente todo o funcionamento da aplicação.

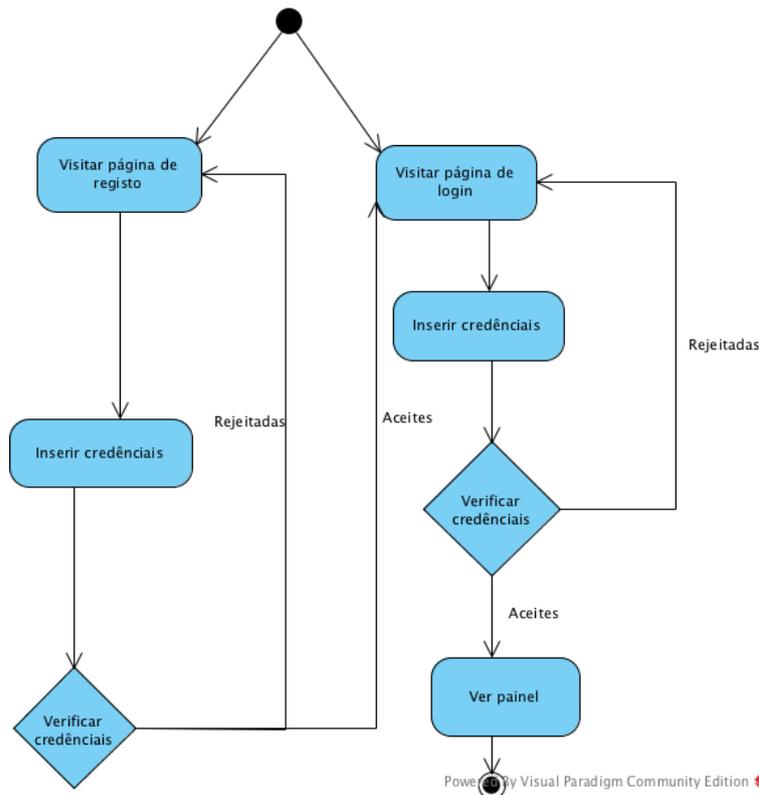


Figura 25 – Diagrama de atividade do processo de registo e autenticação

Na figura 25 é apresentado o comportamento da plataforma perante a autenticação e o registo dos utilizadores. É importante salientar que o registo público está apenas disponível para candidatos, júris e outros administradores são convidados pelo administrador principal através de convites enviados por e-mail. É possível observar que ao chegar à página de registo ou autenticação é obrigatório a inserção das credenciais. Após o registo com sucesso o utilizador é redirecionado para a página de autenticação, caso contrário o sistema mostra a mensagem de erro na página de registo. Se a autenticação é ultrapassada com sucesso, o sistema deve redirecionar o utilizador para o seu painel consoante a função que desempenha na plataforma, caso contrário o utilizador deve permanecer na mesma página.

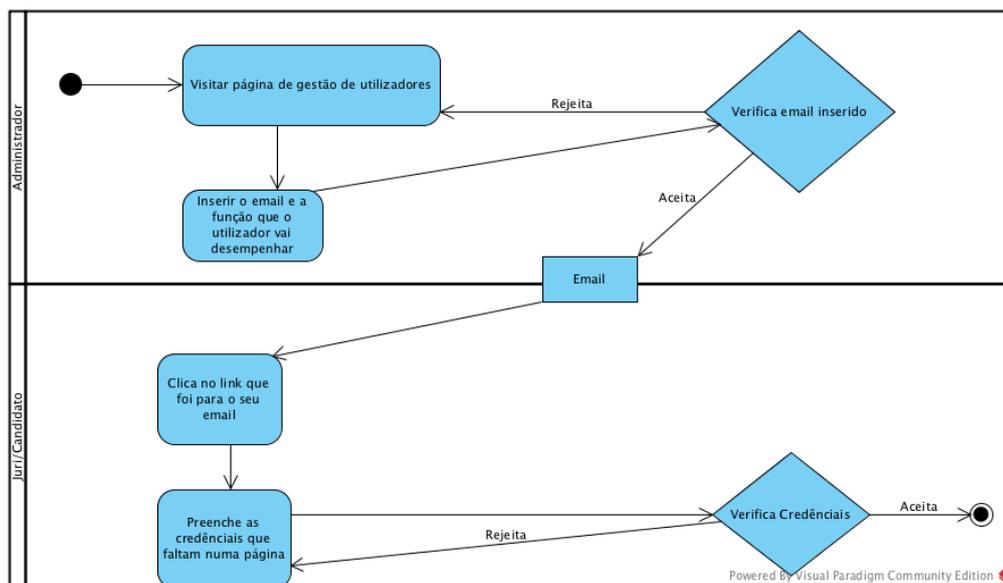


Figura 26 – Diagrama de atividade do processo de convite

A figura 26 representa o modo de funcionamento dos convites que são enviados para júris e outros administradores. Optou-se por este formato de participação uma vez que os júris não podem fazer o seu registo publicamente, mas apenas através de convite enviado pelo administrador.

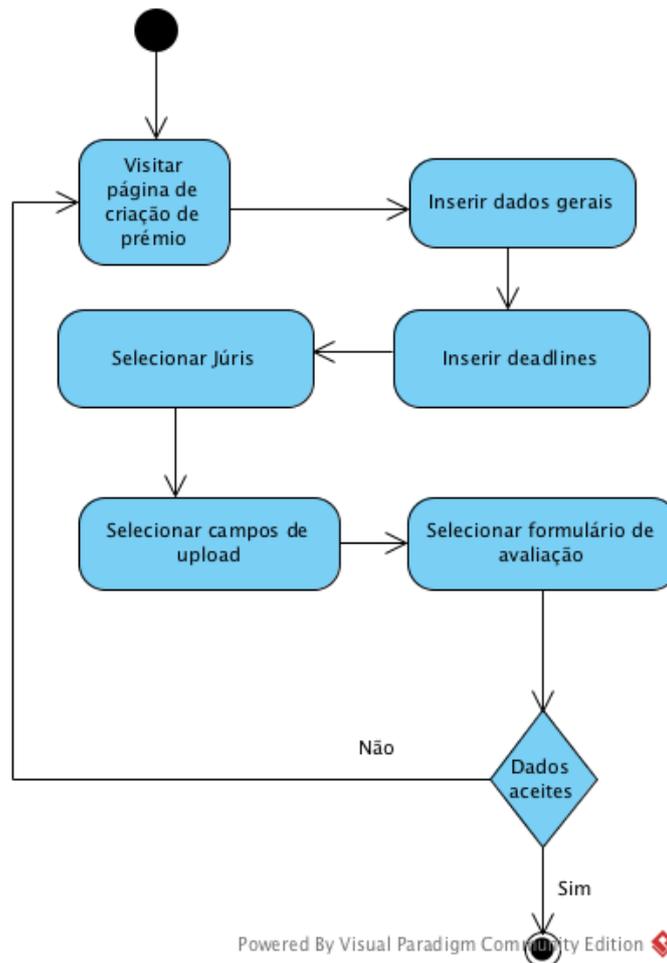


Figura 27 – Diagrama de atividade da adição de prémio

A figura 27 mostra uma das atividades mais relevantes da plataforma, trata-se da possibilidade da criação de um concurso a um prémio por parte dos administradores. Antes de submeter a adição do prémio o administrador deve inserir os dados gerais do prémio, como título e a descrição textual. Depois inserir as datas limite de participação, tanto para envio de candidaturas como de avaliações dessas candidaturas, o administrador tem de selecionar os júris, selecionar os campos de upload que são pedidos quando o candidato preenche a sua candidatura, e decidir se o prémio pedirá na candidatura as publicações em revistas científicas do trabalho que vai a concurso.

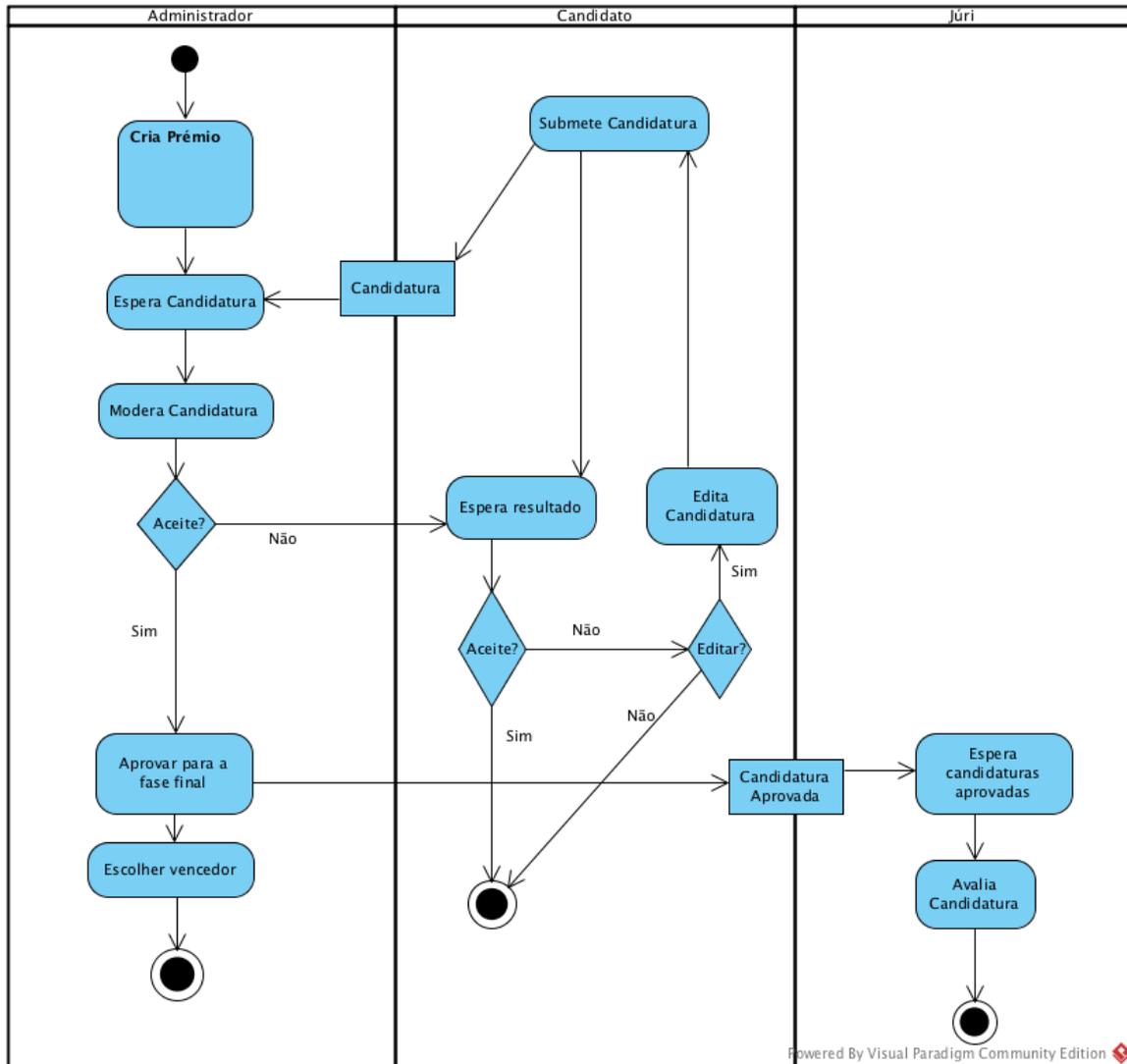


Figura 28 – Processos gerais entre os atores do sistema

A figura 28 apresenta de forma abrangente todas as atividades e ações desempenhadas pelos atores do sistema. O processo começa com a atividade de criação do prémio que pode ser observado na figura 27. Isto revela que se não existirem prémios adicionados na plataforma o candidato e o júri não podem usar nenhuma das suas funcionalidades da aplicação. Depois o administrador fica à espera que sejam submetidas candidaturas ao prémio. Quando o candidato submete a sua candidatura ao prémio em aberto, o administrador fica com as candidaturas disponíveis para moderação. Caso seja aceite esta fica disponível para aprovação para a fase final. Caso seja rejeitada o candidato poderá voltar a editar a candidatura. Para facilitar o processo existe a informação do estado da candidatura no painel. As candidaturas que forem aprovadas para a fase final ficam disponíveis para o júri avaliar até ao prazo inserido pelo administrador nas configurações do prémio. A participação do candidato termina quando a sua candidatura é aceite pelo administrador, o júri depois de fazer a avaliação às candidaturas que entender e o administrador depois de selecionar o vencedor do concurso. Os concursos aceitam mais do que um vencedor.

4.3.2. Vista de desenvolvimento

Esta vista é usada na descrição da arquitetura, uma vez que apresenta o sistema na perspectiva do programador. A finalidade é mostrar a organização dos módulos de software

e detalhes da implementação. Para construir a vista foi usado um diagrama de pacotes, onde cada componente representa um módulo do software desenvolvido.

Na figura 29 são apresentados os ficheiros e pastas constituintes da aplicação desenvolvida. É composta por quatro componentes principais: core, controller, model e view. O core contém as classes gerais relacionadas com encriptação, acesso à base de dados, autenticação, entre outras. O controller possui os ficheiros relativos aos controladores que são executados no formato “/controlador/parâmetro/” através do URL. O model contém as classes relativas à lógica funcional da aplicação. É com estas classes que também se assegura a utilização da base de dados. A view contém os ficheiros que compõem o front-end da aplicação e dentro dela estão três pastas relativas ao painel do júri, administrador e candidato.

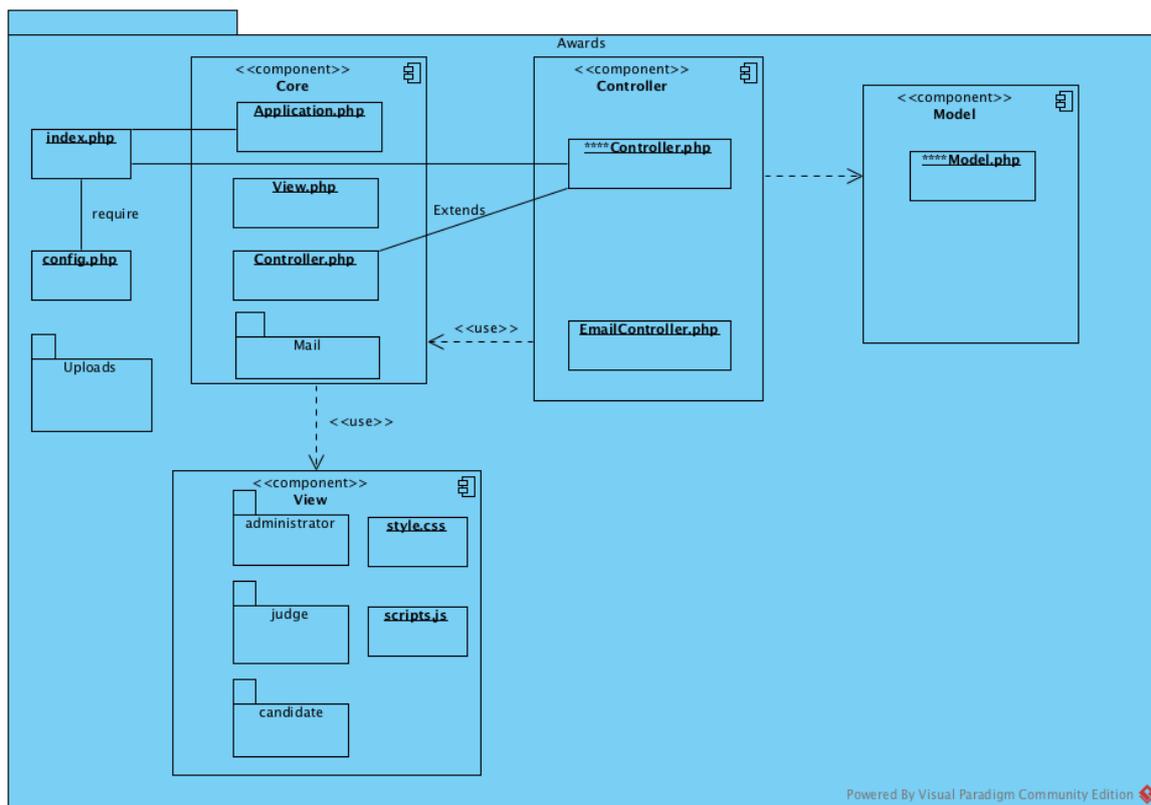


Figura 29 – Vista de desenvolvimento

O funcionamento da aplicação começa com o carregamento das classes do core, controller e model através do ficheiro “index.php” que se encontra na pasta principal da aplicação. Este ficheiro chama a parte do core encarregue da gestão dos controladores executados através do URL, a classe “Application” visível na figura X. Os controladores invocam funções das classes pertencentes ao model e por isso existe uma relação de dependência entre os componentes e também usam a classe “View.php” do core, para gerar a página pedida pelo utilizador. A biblioteca Swift Mailer encontra-se dentro da pasta do core e é usada pelo controlador.

O core é responsável por tarefas gerais que asseguram o correto funcionamento da aplicação. A tabela 18 mostra a responsabilidade de cada classe.

Classe	Responsabilidade
Application	Classe responsável pela inicialização do controlador passado através do URL com possibilidade da utilização de parâmetros.

Auth	Verifica se um determinado controlador está a ser executado por um utilizador registado e com permissão.
Controller	Classe-mãe de todas as classes pertencentes à diretoria “controller”.
Database	Trata da ligação com a base de dados.
Encryption	Oferece funções de encriptação e descriptação necessárias para o funcionamento da aplicação.
Filter	Proteção contra ataques cross-site scripting.
Request	Responsável por encapsular o acesso às variáveis globais relacionadas com a inserção de dados.
Session	Encapsulamento do acesso à variável global de sessão.
Token	Proteção contra ataques cross-site request forgery.
View	Classe que gera a página pedida por determinado controlador.

Tabela 18 – Classes do core da aplicação

As classes contidas no model possuem funções relacionadas com a lógica de negócio de prémios, candidaturas, avaliações, envios de e-mails, opções da plataforma, etc. É através delas que é feito o acesso à informação contida na base de dados.

O controller tem classes que gerem o fluxo da aplicação. As funções dessas classes são responsáveis por chamar a função da classe “View” do core para gerar a página com informação retornada por funções do model. Todas elas descendem da classe “Controller” que está no core. Cada controlador antes de ser chamado garante que o utilizador tem privilégios para o executar.

4.3.3. Vista física

Esta vista mostra como os elementos de software se integram com o hardware, oferecendo assim uma perspectiva da topologia dos elementos que constituem a arquitetura.

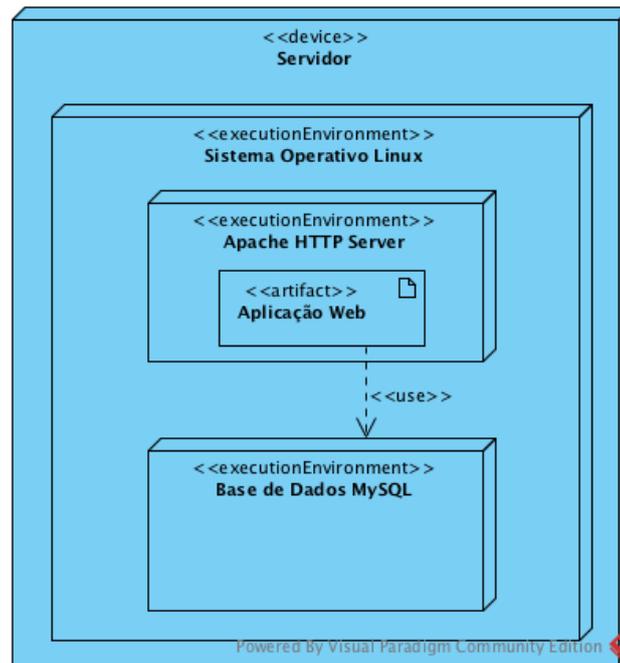


Figura 30 – Vista física

Na figura 30 não são observadas ligações, uma vez que todos os componentes da solução são executados na mesma máquina.

4.4. Modelo da base de dados

Nesta secção é apresentada a base de dados desenvolvida para o projeto. Esta possui dezassete tabelas necessárias para o funcionamento da plataforma.

Em PHP para aceder à base de dados MySQL existem duas possibilidades: MySQLi ou PHP Data Objects. Apesar de disponibilizarem ambos uma API orientada a objetos, o desempenho do PDO é superior em alguns casos^[41]. É importante salientar que o escolhido foi o PDO devido à portabilidade que apresenta em relação ao concorrente, que apenas serve se o motor de base de dados for MySQL. Com a utilização do PDO, futuramente será possível^[42] alterar o motor sem grandes custos.

Na figura 31 é apresentado o esquema da base de dados contendo as relações entre tabelas e atributos de cada uma delas.

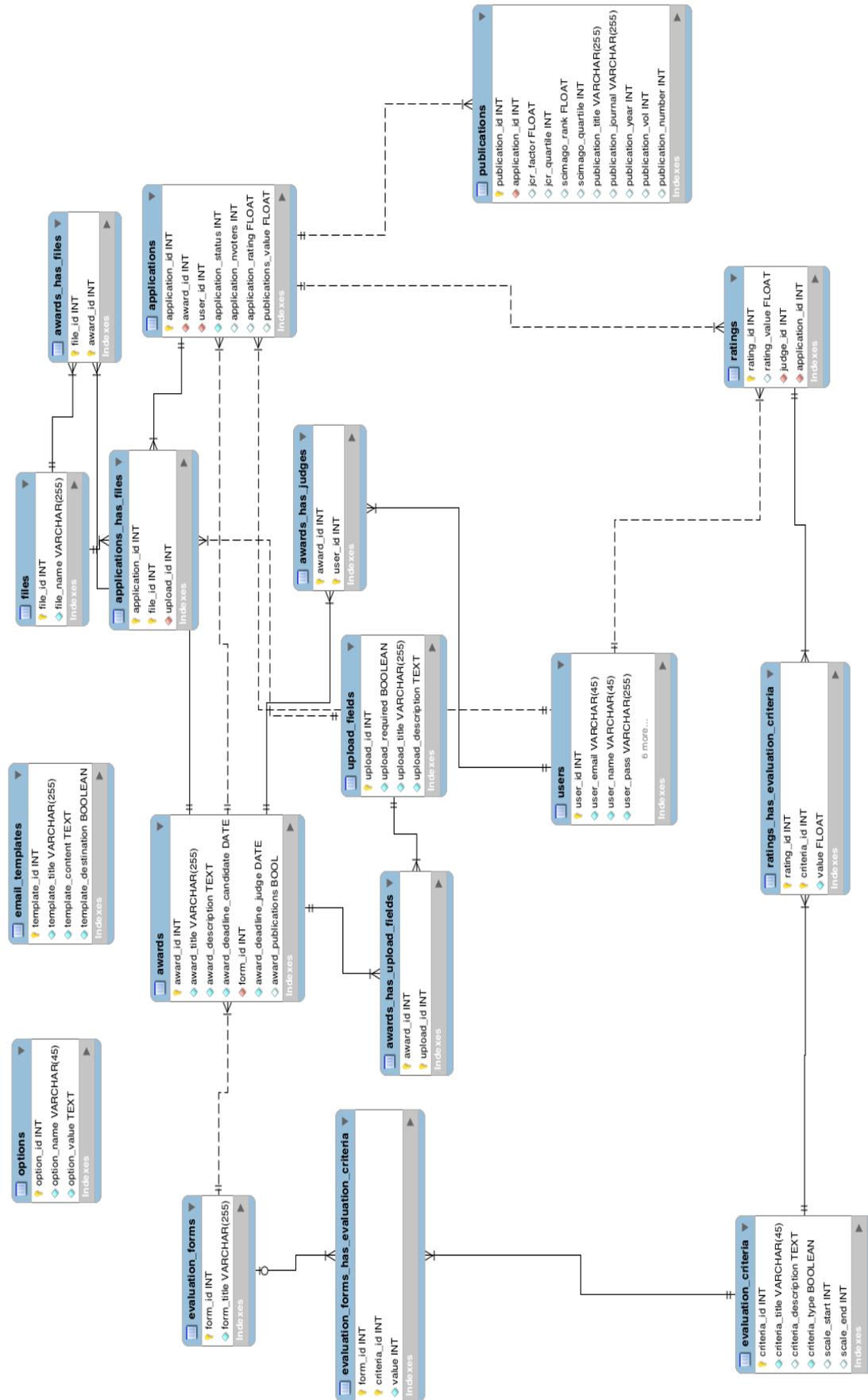


Figura 31 – Modelo da base de dados

De seguida é apresentada uma descrição das tabelas e atributos relevantes que são necessários para o funcionamento da plataforma.

- **Awards** – Esta tabela armazena os dados relativos ao prémio, tais como: id, título, descrição, datas limite de candidatura e avaliação, o id do formulário de critérios de avaliação, e se as candidaturas devem conter as publicações.
- **Awards has judges** – Relaciona os prémios com os júris selecionados para esse prémio. Possui como atributos o id do júri e o prémio a que este foi associado.
- **Awards has upload fields** – É responsável pelo armazenamento dos campos de upload pedidos ao candidato na sua candidatura. Como atributos tem o id do campo de upload e o id do prémio.
- **Awards has files** – Ficheiros adicionados na tabela Files relacionados com o regulamento do prémio.
- **Upload fields** – Encarregue de guardar os uploads necessários para a candidatura de determinado prémio. Possui um id, título, descrição e um atributo para classificar o campo de upload como obrigatório.
- **Applications** – Informações relativas à candidatura como: id, prémio, utilizador, estado, número de avaliadores, valor da avaliação e das publicações.
- **Applications has files** – Ficheiros adicionados na tabela Files relacionados com as submissões feitas aos campos de upload pedidos pelo prémio
- **Files** – Tabela com o id e nome do ficheiro submetido na plataforma.
- **Email templates** – Armazena os templates utilizados no envio de mensagens para vários utilizadores simultaneamente.
- **Evaluation forms** – Esta tabela armazena o título e id de um formulário de critérios de avaliação.
- **Evaluation criteria** – Responsável pelo armazenamento dos critérios de avaliação criados por administrador. Possui atributos como o id, título, descrição, tipo de critério e onde começa e acaba a escala de votação para o critério.
- **Evaluation forms has criteria** – Esta tabela relaciona determinado formulário com os critérios que o compõem.
- **Users** – Armazena todas as informações relevantes do utilizador, tais como: id, email, nome, senha, função, estado, etc.
- **Publications** – As candidaturas submetidas pelo utilizador podem conter publicações do trabalho caso o administrador assim o indicar. Aqui são guardadas as informações de cada publicação.
- **Ratings** – As avaliações atribuídas às candidaturas são guardadas através desta tabela, que contém atributos como: nota da avaliação, id de um júri, e a aplicação a que se refere.
- **Ratings has criteria** – Aqui são guardados os valores atribuídos a cada critério na votação que o júri faz.
- **Options** – Armazena valores necessários para as configurações gerais da plataforma.

4.5. Segurança

A segurança foi tida em conta no desenvolvimento da aplicação, e por isso foram analisadas várias vulnerabilidades comuns^[43] neste estilo de plataforma. De seguida são descritos os mecanismos usados para lidar com estes problemas.

Encriptação dados de sensíveis

Para proteger a senha do utilizador na base de dados é criada uma hash da senha através da função “password_hash()” do PHP. Esta função utiliza^[44] o algoritmo bcrypt e é recomendado que o atributo da tabela destinado à senha possua 255 caracteres. Desta forma se a base de dados for comprometida as senhas não estão legíveis.

Utilização de HTTPS

A utilização do HTTPS é importante uma vez que este adiciona uma camada de segurança ao protocolo HTTP. Em conjunto com o protocolo SSL/TLS protege o tráfego transmitido por meio de uma ligação encriptada que permite verificar a autenticidade do servidor e do cliente através de certificados digitais.

Proteção contra acesso a ficheiros

Uma vez que a aplicação se divide em diretorias com ficheiros, é importante restringir o acesso direto aos ficheiros através do front-end. Para lidar com este problema foi criado um ficheiro “.htaccess” nas diretorias core, model e controller que proíbe qualquer tipo de acesso aos ficheiros de determinada diretoria.

SQL Injection

O SQL injection é uma ataque malicioso em que o atacante executa instruções SQL para se aproveitar de sistemas que interagem com base de dados. No caso desta aplicação foram usados “prepared statements” em todas as queries, como forma de prevenir^[45] este tipo de ataques.

Cross-site scripting

O cross-site scripting (XSS) é um conhecido^[46] ataque em aplicações web, no qual o atacante injeta código Javascript num campo de texto já existente, e que fica disponível para outros utilizadores que acederem à página. Para se lidar com este problema, foi usada uma função^[47] nativa do PHP que lida com esta vulnerabilidade. O seu propósito é converter caracteres especiais em entidades HTML.

Cross-site request forgery

É um tipo de ataque malicioso em que pedidos não autorizados são transmitidos através de um utilizador em quem a aplicação confia. O CSRF^[48] explora a confiança que uma aplicação tem no browser que está a ser utilizado. Como forma de garantir a proteção contra este tipo de ataque foi criado um método que é chamado cada vez que são submetidos dados em formulários. Nas funções dos controladores, o token é verificado para garantir que o pedido foi mesmo feito pelo browser do utilizador.

Cookie

Em relação à segurança do cookie, este foi configurado para ser acessível apenas através do protocolo HTTP. Isto significa que o cookie não será acessível por linguagens de script, como Javascript. Esta configuração^[49] pode efetivamente ajudar a reduzir o roubo de identidade por meio de ataques XSS. Para além disto a informação contida no cookie também se encontra toda encriptada para não ser facilmente alterada por um utilizador mal intencionado.

Brute Force e Spam

Outro aspecto que foi tido em conta são os spam bots. Para a proteção de formulários públicos como o registo e o login foi usado um serviço grátis reCAPTCHA^[50]

da Google. No entanto para estar funcional é necessário o fornecimento do endereço público onde é instalada a aplicação.

Capítulo 5

5. Testes

No desenvolvimento de software é de extrema relevância testar o produto. Deve ser garantido que a aplicação cumpre os objetivos definidos pelos requisitos. De forma a validar o sistema desenvolvido foram definidos dois tipos de testes: funcionais e de usabilidade. Nesta secção encontra-se toda a informação relevante sobre os testes realizados.

5.1 Testes funcionais

Depois de terminada a implementação de um projeto, é importante realizar testes funcionais para garantir a qualidade do produto e que todas as funcionalidades definidas pelos requisitos estão implementadas e de acordo com o esperado. A tabela 19 mostra o exemplo do registo de um teste funcional.

ID	F.1.1
Nome	Registar
Descrição	O candidato deve poder registar-se através do formulário público disponível na página de registo para poder usar a plataforma.
Pré-condições	<ul style="list-style-type: none"> • Inserir um email único • Inserir o nome • Inserir uma senha
Resultado esperado	O utilizador é inserido na base de dados e o sistema envia um email para confirmação do registo.
Resultado	Sucesso

Tabela 19 – Exemplo de um teste funcional

As tabelas 20, 21, 22 e 23 mostram os testes realizados às funcionalidades do sistema.

Utilizador

ID	Requisito	Prioridade	Testado
F.1.1	Registar	Must	Sim
F.1.2	Login	Must	Sim
F.1.3	Editar perfil	Must	Sim
F.1.4	Logout	Must	Sim
F.1.5	Sessão expirar	Should	Não
F.1.6	Confirmar registo	Must	Sim
F.1.7	Finalizar registo pelo convite	Must	Sim

F.1.8	Mostrar últimos prémios	Should	Sim
-------	-------------------------	--------	-----

Tabela 20 – Testes funcionais do utilizador

Administrador

ID	Requisito	Prioridade	Testado
F.2.1.1	Editar configurações da plataforma	Should	Sim
F.2.2.1	Convidar um membro	Must	Sim
F.2.2.2	Mostrar lista de utilizadores registados	Must	Sim
F.2.2.3	Editar utilizador	Must	Sim
F.2.2.4	Remover utilizador	Must	Sim
F.2.3.1	Mostrar lista de templates de mensagem	Must	Sim
F.2.3.2	Adicionar template de mensagem	Must	Sim
F.2.3.3	Remover template de mensagem	Must	Sim
F.2.3.4	Editar template de mensagem	Must	Sim
F.2.4.1	Enviar mensagem para júris ou candidatos de um prémio	Must	Sim
F.2.5.1	Adicionar formulário de critérios de avaliação	Must	Sim
F.2.5.2	Remover formulário de critérios de avaliação	Must	Sim
F.2.5.3	Editar formulário de critérios de avaliação	Must	Sim
F.2.5.4	Mostrar lista de formulários de critérios de avaliação	Must	Sim
F.2.6.1	Adicionar critério de avaliação	Must	Sim
F.2.6.2	Editar critério de avaliação	Must	Sim
F.2.6.3	Remover critério de avaliação	Must	Sim
F.2.6.4	Mostrar lista de critérios de avaliação	Must	Sim
F.2.7.1	Adicionar campo de upload	Must	Sim
F.2.7.2	Remover campo de upload	Must	Sim
F.2.7.3	Editar campo de upload	Must	Sim
F.2.7.4	Mostrar lista de campos de upload	Must	Sim
F.2.8.1	Adicionar prémio	Must	Sim

F.2.8.2	Remover prémio	Must	Não
F.2.8.3	Editar prémio	Must	Sim
F.2.8.4	Mostrar lista de prémios	Must	Sim
F.2.9.1	Moderar candidaturas pendentes	Must	Sim
F.2.9.2	Enviar candidatura para avaliação	Must	Sim
F.2.9.3	Ordenar candidaturas aprovadas	Must	Sim
F.2.9.4	Escolher candidatura vencedora	Must	Sim
F.2.9.5	Remover candidatura	Must	Não

Tabela 21 – Testes funcionais do painel do administrador

Júri

ID	Requisito	Prioridade	Testado
F.3.1.1	Mostrar histórico de prémios	Should	Sim
F.3.2.1	Mostrar candidaturas para avaliação	Must	Sim
F.3.3.1	Mostrar ficheiros da candidatura	Must	Sim
F.3.4.1	Mostrar lista de avaliações	Must	Sim
F.3.5.1	Remover avaliação da candidatura	Must	Não
F.3.6.1	Editar avaliação de candidatura	Must	Sim

Tabela 22 – Testes funcionais do painel do júri

Candidato

ID	Requisito	Prioridade	Testado
F.4.1.1	Submeter candidatura	Must	Sim
F.4.2.1	Mostrar candidaturas	Must	Sim
F.4.3.1	Mostrar estado da candidatura	Must	Sim
F.4.4.1	Remover candidatura	Should	Não
F.4.5.1	Editar candidatura	Must	Sim

Tabela 23 – Testes funcionais do painel do candidato

A tabela criada para cada teste pode ser consultada no **Anexo D – Testes**. Existem testes que não foram efetuados pois as funcionalidades ainda demonstravam imperfeições quando foram realizados os testes.

5.2 Testes de usabilidade

Os testes de usabilidade são importantes no contexto do desenvolvimento de uma aplicação web uma vez que permitem analisar a interação dos utilizadores com o sistema. A

finalidade é determinar se a aplicação é facilmente utilizada pelos utilizadores, e em caso contrário perceber como se pode melhorar o front-end da aplicação.

Para obter resultados dos testes de usabilidade, no inquérito os testers foram questionados sobre o desempenho da aplicação em relação a critérios como o feedback, linguagem e navegabilidade do sistema. Estes critérios foram definidos tendo em conta as dez heurísticas propostas por Nielsen^[51] sobre o design de interfaces. Para o teste foram escolhidas tarefas de cada um dos três tipos de utilizadores da plataforma. Foi pedido ao tester que avaliasse a facilidade de aprendizagem e sucesso da execução das tarefas. Apesar do teste ter sido presencial não houve interação direta entre o desenvolvedor e o tester e os resultados foram recolhidos através de um inquérito online. A tabela 24 mostra as tarefas que os testers tiveram de executar.

Tarefa	Administrador	Júri	Candidato
1	Login	Login	Login
2	Criar formulário com critérios já existentes	Submeter avaliação a candidatura	Submeter candidatura a um prémio em aberto
3	Criar Prémio	Logout	Logout
4	Adicionar template de mensagem		
5	Selecionar destinatários para envio de uma mensagem através de template		
6	Moderar candidaturas já existentes		
7	Editar título da plataforma		
8	Logout		

Tabela 24 – Tarefas dos testes de usabilidade

Depois de recolhidos os dados dos inquéritos, foram analisados e foi possível constatar que as tarefas foram todas executadas com sucesso. A figura 32 apresenta um gráfico com a média de votações do critério de facilidade de aprendizagem em cada tarefa desempenhada pelo tester.

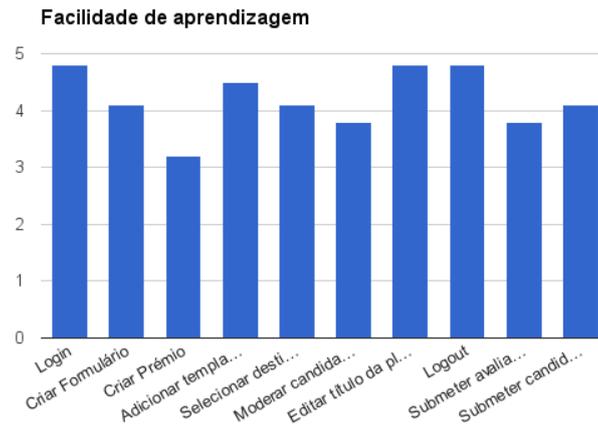


Figura 32 – Facilidade de aprendizagem do sistema

Ao observar o gráfico é possível verificar que a tarefa de criação de um prémio foi a que provocou mais dificuldades aos testers. Como a adição de um prémio precisa de um formulário de critérios de avaliação e de campos de upload para funcionar tornou o processo um pouco mais complexo para o tester. A ideia principal retida da visualização do decorrer dos testes foi que o processo de adição do prémio deve ser realizado por inteiro na mesma página, os campos de upload possuem um próprio menu, no entanto devem poder ser introduzidos pela página da adição de prémio também. Todas as outras tarefas possuem uma pontuação alta porque são extremamente simples de identificar e executar.

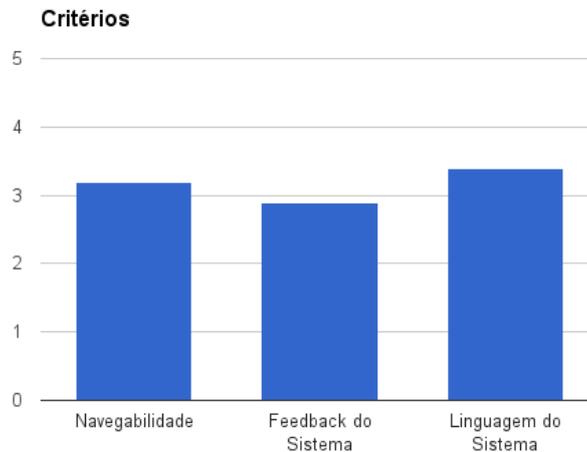


Figura 33 – Critérios do inquérito dos testes de usabilidade

Na figura 33 são apresentadas as pontuações referentes à navegabilidade, feedback e linguagem do sistema. Estes resultados indicaram que havia a necessidade de reformular as mensagens de erro do sistema e introduzir algumas inexistentes. Em termos de navegabilidade foram implementados breadcrumbs^[52] que melhoram a navegação na ramificação dos menus. Importa salientar que nos testes de usabilidade participaram doze pessoas relacionadas com o meio científico. Apesar das tarefas não serem complexas conseguiu-se identificar alguns problemas relacionados com o front-end da aplicação.

Capítulo 6

6. Resultados finais

Este capítulo tem como objetivo mostrar os resultados conseguidos depois da implementação. Serão mostradas imagens das janelas mais relevantes do funcionamento da aplicação.



Figura 34 – Página principal da aplicação

Na figura 34 é apresentada o formato da página principal, onde são listados os últimos concursos geridos na plataforma.

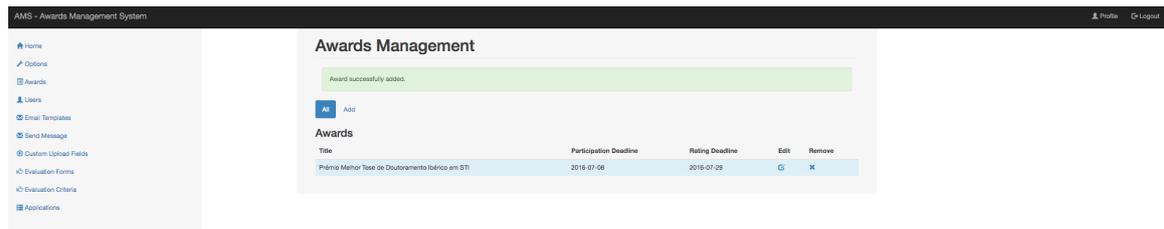


Figura 35 – Mensagem de aviso ao utilizador

Acima é possível observar uma mensagem de sucesso do sistema. O feedback das ações executadas é mostrado naquele formato em todas as páginas da plataforma.

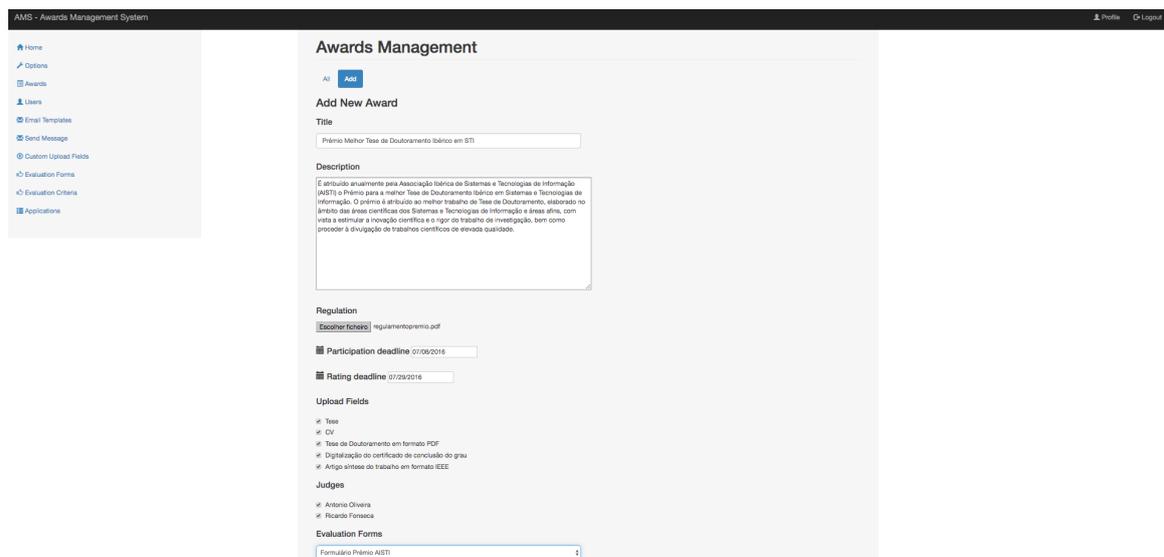


Figura 36 – Adicionar prémio

Na figura 36 é possível observar o menu de criação de um prémio científico. Trata-se de uma das funcionalidades fulcrais para o funcionamento de toda a plataforma.

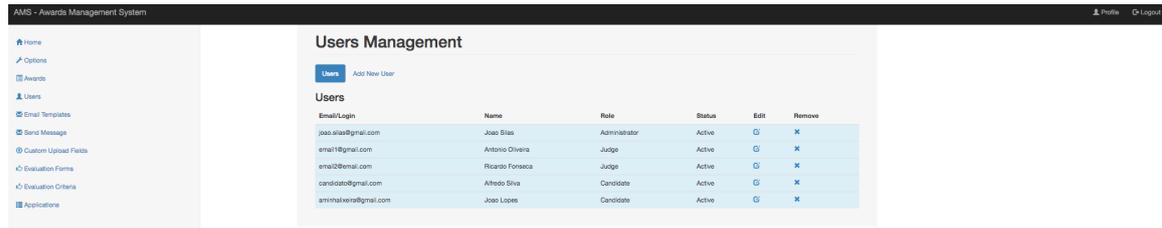


Figura 37 – Gestão de utilizadores da plataforma

A figura 37 apresenta a lista de utilizadores do sistema. A partir daqui é possível remover, editar informações, e o estado de um determinado utilizador.



Figura 38 – Candidaturas aprovadas

É neste menu com as candidaturas aprovadas que o administrador as pode seleccionar para a fase final do concurso, onde vão ser avaliadas pelo júri.

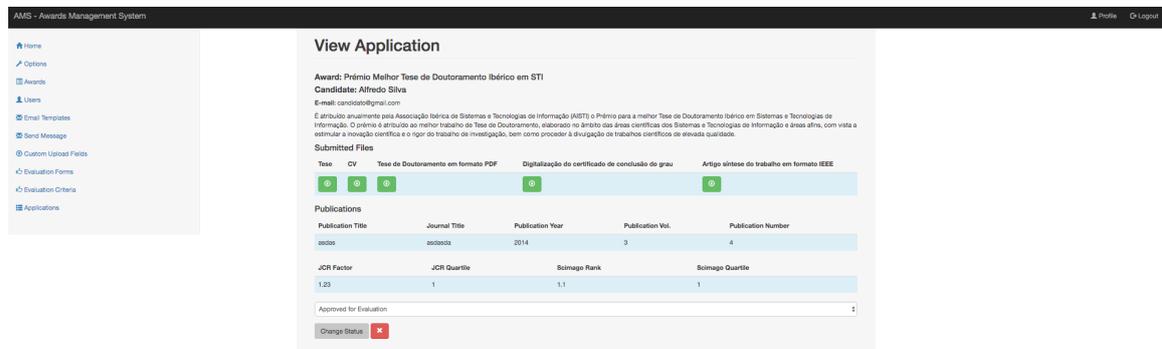


Figura 39 – Página com a informação da candidatura

A figura 39 apresenta a página com a informação total relacionada a uma determinada candidatura.

Submit Application

Prémio Melhor Tese de Doutoramento Ibérico em STI

Description

É atribuído anualmente pela Associação Ibérica de Sistemas e Tecnologias de Informação (ASTI) o Prémio para a melhor Tese de Doutoramento Ibérico em Sistemas e Tecnologias de Informação. O prémio é atribuído ao melhor trabalho de Tese de Doutoramento, elaborado no âmbito das áreas científicas dos Sistemas e Tecnologias de Informação e áreas afins, com vista a estimular a inovação científica e o rigor do trabalho de investigação, bem como proceder à divulgação de trabalhos científicos de elevada qualidade.

Deadline

07/08/2016 - (11:58pm)

Files

(*) Required file.

Tese (*)

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Escolher ficheiro Nenhum ficheiro selecionado

CV (*)

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Escolher ficheiro Nenhum ficheiro selecionado

Tese de Doutoramento em formato PDF (*)

Tese de Doutoramento em formato PDF

Escolher ficheiro Nenhum ficheiro selecionado

Digitalização do certificado de conclusão do grau (*)

Digitalização do certificado de conclusão do grau

Escolher ficheiro Nenhum ficheiro selecionado

Figura 40 – Submissão de candidatura

Na figura 40 é apresentado o formulário de submissão de candidatura com as informações relativas ao prémio e aos campos de upload pedidos.

Submitted Files

Tese	CV	Tese de Doutoramento em formato PDF	Digitalização do certificado de conclusão do grau	Artigo síntese do trabalho em formato IEEE

Publications

Value: 21.35

Title	Journal	Year	Vol.	N°
2008	2014	3	4	

JCR Factor	JCR Quartile	SCImago Rank	SCImago Quartile
1.23	Q1	1.1	Q1

Evaluation

Metodologia

Avalia-se a adequabilidade do desenho de investigação selecionado ao problema de investigação objecto da tese.

Rate:

Apresentação dos Resultados

Avalia-se a forma como os resultados são apresentados, e.g., a adequabilidade quanto ao nível e a forma de análise tendo em conta os dados obtidos.

Rate:

Revisão da Literatura

A tese revela a utilização alargada e crítica de uma gama vasta de teorias e de fontes de informação.

Rate:

Figura 41 – Painel de avaliação da candidatura

Na figura 41 é apresentado o formulário de avaliação de candidatura que é preenchido pelo júri.

Capítulo 7

7. Conclusões e trabalho futuro

O primeiro semestre começou com a definição das tarefas a fazer ao longo do semestre. Logo de seguida foi feito o enquadramento no contexto do problema que culminou na escrita do estado da arte. As atividades envolvidas no estudo do estado da arte foram estimadas para três semanas, no entanto não foram suficientes para terminar a escrita de todo o conteúdo. Este capítulo sofreu uma profunda alteração no segundo semestre, já que um dos softwares principais do estudo foi descontinuado pelo seu autor. Na maioria dos softwares analisados não foi possível a disponibilização de demonstrações e isto condicionou a análise do contexto e funcionamento concreto das funcionalidades.

De seguida foram iniciadas as atividades de análise de requisitos: definição e caracterização de utilizadores, análise de processos, criação de user stories e por fim a definição dos requisitos recorrendo-se a casos de uso. Para fazer uso desta técnica foram necessárias algumas reuniões com o cliente para alguns processos serem mais facilmente entendidos e registados.

O objetivo principal para o primeiro semestre foi criar um protótipo não-funcional da plataforma. Para isso recorreu-se ao desenvolvimento de mockups para as duas versões de protótipos planeadas. Os mockups foram o principal instrumento na discussão das funcionalidades com o cliente. Apesar do tempo de criação de um dos protótipos não ter sido respeitado, o objetivo final cumpriu-se uma vez que foi validado.

Inicialmente estava previsto que o estudo da arquitetura geral da plataforma fosse antes do desenvolvimento dos mockups. Isto não aconteceu, uma vez que o cliente identificou algumas funcionalidades em falta e por isso o tempo de atividade do registo dos requisitos aumentou.

O estudo da arquitetura geral foi feito na última semana de dezembro e no início do segundo semestre, durante esse tempo foram revistos aspectos relacionados com o modelo da base de dados e segurança.

Antes do segundo semestre ter início foram feitas alterações no plano, no entanto a implementação da parte do painel do administrador sofreu um atraso. Este atraso foi provocado pela má estimativa do tempo de desenvolvimento das funcionalidades. Apesar do plano de mitigação para este risco apontar para a divisão da atividade em atividades mais pequenas, não foi possível cumprir o prazo das quatro semanas, tendo o desenvolvimento deste painel sido sobreposto com tarefas de desenvolvimento dos outros painéis. Como isto sucedeu foi impossível cumprir as quatro semanas de testes que eram expectáveis, e por isso os testes foram feitos ao longo de três semanas. Apenas foram efetuados dois tipos de testes e em ambos os casos foram conseguidas melhorias no software.

Os objetivos gerais foram alcançados uma vez que a aplicação resultante do estágio se encontra testada e validada pelo cliente. Esta cumpre os requisitos mais importantes na totalidade.

O desenvolvimento desta aplicação foi importante para o estagiário porque adquiriu conhecimentos de uma linguagem importante no desenvolvimento de aplicações web. O trabalho futuro passa por efetuar testes em ambiente real, e implementação das técnicas de combate ao spam bots, que necessitam de um domínio público para funcionarem. Além disto é necessário melhorar a documentação do código, e implementar mais melhorias no front-end da plataforma.

Referências

1. IEEE International Conference on Robotics and Automation. (2016, Março 18). Retrieved from <http://www.icra2016.org/>
2. IEEE International Conference on Communications. (2016, Março 18). Retrieved from <http://icc2016.ieee-icc.org/>
3. IEEE International Conference On Accoustics, Speech and Signal Processing. (2016, Março 18). Retrieved from <http://www.icassp2016.org/>
4. Regulamento Prémio Melhor Tese de Doutoramento Ibérico em STI. (2015, Setembro 18). Retrieved from <http://www.aisti.eu/regulamentopremio.pdf>
5. Regulamento Prémio Melhor Artigo Científico. (2015, Setembro 18). Retrieved from <http://www.uc.pt/fmuc/gai/premios/premiosgai2014/PremioArtigo/regulamentoa rtigo>
6. Prémio Científico IBM. (2015, Setembro 18). Retrieved from <http://www-05.ibm.com/pt/pc/premio.html>
7. Prémio Científico Mário Quartin Graça. (2015, Setembro 18). Retrieved from <http://casamericalatina.pt/premios/premio-cientifico/>
8. Prémio Científico Sesimbra. (2015, Setembro 18). Retrieved from <http://gadi.ufp.pt/premio-cientifico-sesimbra/>
9. Prémios Científicos ISCTE-IUL. (2015, Setembro 18). Retrieved from http://www.iscte-iul.pt/servicos/gabinete_de_apoio_a_investigacao/scientific_awards.aspx
10. Pierskalla Best Paper Award. (2016, Março 11). Retrieved from <https://www.informs.org/Recognize-Excellence/Community-Prizes-and-Awards/Health-Applications-Society/Pierskalla-Best-Paper-Award>
11. CSS Junior Scientific Award. (2016, Março 11). Retrieved from http://www.cpt.univ-mrs.fr/~barrat/regulation_junior_award.pdf
12. Dijkstra Prize. (2016, Março 11). Retrieved from http://www.iscte-iul.pt/servicos/gabinete_de_apoio_a_investigacao/scientific_awards.aspx
13. Eppendorf Award for Young European Investigators. (2016, Março 11). Retrieved from <http://corporate.eppendorf.com/en/company/scientific-awards/european-award/>
14. ICRA Cognitive Robotics Best Paper Award. (2016, Março 11). Retrieved from <http://www.ieee-ras.org/awards-recognition/conference-awards/69-awards-recognition/society-awards/83-icra-cognitive-robotics-best-paper-award-sponsored-by-cotesys>
15. Wizehive | Enterprise collaboration, business process, and workflow. (2016, Março 11). Retrieved from <http://www.wizehive.com/>
16. Judgify: Awards judging system. (2016, Março 11). Retrieved from <https://www.judgify.me>
17. OpenWater – Online Award Judging Software. (2016, Março 11). Retrieved from <http://www.getopenwater.com/award-management-software>
18. OpenWater | Online Submissions. (2016, Março 11). Retrieved from <http://www.getopenwater.com/online-award-submission-software>
19. Smarter Select – Online Application System. (2016, Março 11). Retrieved from <http://www.smarterselect.com/>
20. FluidReview - Colect and Review Applications. (2016, Março 11). Retrieved from <https://fluidreview.com/>
21. Ex Ordo – All-in-One Abstract Management Software. (2016, Março 11). Retrieved from <https://www.exordo.com/>

22. OpenConf | Peer-Review, Abstract and Conference Management. (2015, Setembro 18). Retrieved from <https://www.openconf.com/>
23. EasyChair. (2016, Março 11). Retrieved from <http://www.easychair.org/>
24. SmartConfoo – Your Conference Management Software. (2016, Março 11). Retrieved from <http://iris-solutions.ca/en/products/smartconfoo/>
25. Conference Management System. (2016, Março 11). Retrieved from <http://www.conference-service.com/conference-support/conference-management-system.html>
26. LimeSurvey – The most popular FOSS survey tool on the web. (2015, Setembro 18). Retrieved from <http://iris-solutions.ca/en/products/smartconfoo/>
27. Prémio Melhor Artigo Científico do Gabinete de Apoio à Investigação da FMUC. (2015, Setembro 18). Retrieved from <http://www.uc.pt/fmuc/gai/premios/premiosgai2014/PremioArtigo/regulamentoa rtigo>
28. MoSCoW Prioritization. (2015, Outubro 20). Retrieved from <http://www.bawiki.com/wiki/techniques/moscow-prioritization/>
29. Otero, C., 2012. Software Engineering Design: Theory and Practice. 1st ed. New York: CRC Press
30. Sweat, J., 2005. php | architect's Guide to PHP Design Patterns. 1st ed. Canada: Marco Tabini & Associates, Inc.
31. Zend Framework. (2015, Dezembro 27). Retrieved from <https://framework.zend.com/manual/1.12/en/learning.quickstart.intro.html>
32. CodeIgniter User Guide. (2015, Dezembro 27). Retrieved from http://www.codeigniter.com/user_guide/
33. CakePHP Book. (2015, Dezembro 27). Retrieved from <http://book.cakephp.org/3.0/en/index.html>
34. Laravel Documentation. (2015, Setembro 29). Retrieved from <https://laravel.com/docs/5.2>
35. Pros and Cons of most Favoured PHP Frameworks. (2015, Outubro 04). Retrieved from <https://www.agriya.com/blog/2015/08/31/pros-cons-favoured-php-frameworks/>
36. Bootstrap - Components. (2016, Fevereiro 25). Retrieved from <http://getbootstrap.com/components/>
37. Bootstrap - Examples. (2016, Fevereiro 25). Retrieved from <http://v4-alpha.getbootstrap.com/examples/>
38. Swift Mailer - Book. (2016, Maio 25). Retrieved from <http://swiftmailer.org/docs/introduction.html>
39. Clements, P., 2001. Documenting Software Architectures. 2nd ed. United States: Addison-Wesley.
40. Kruchten, P., 1995. The “4+1” Model of Software Architecture.
41. PDO vs MySQLi performance comparison. (2016, Março 5). Retrieved from <http://wooptoo.com/blog/pdo-vs-mysqli-performance-comparison/>
42. Why you should use PDO instead MySQLi. (2016, Março 5). Retrieved from <https://www.conetix.com.au/blog/why-you-should-use-pdo-instead-mysqli>
43. Serrão, C., Marques, J., 2009. Programação com PHP 5.3. 1nd ed. Lisboa: FCA.
44. PHP – Documentation. (2016, Fevereiro 10). Retrieved from <http://php.net/manual/en/function.password-hash.php>
45. How Can I Prevent SQL Injection. (2016, Fevereiro 10). Retrieved from <http://stackoverflow.com/questions/60174/how-can-i-prevent-sql-injection-in-php/60496#60496>
46. Cross-site scripting. (2016, Fevereiro 10). Retrieved from https://en.wikipedia.org/wiki/Cross-site_scripting

47. PHP - Documentation. (2016, Abril 21). Retrieved from <http://php.net/manual/en/function.htmlspecialchars.php>
48. Cross-site request forgery. (2016, Abril 21). Retrieved from https://pt.wikipedia.org/wiki/Cross-site_request_forgery
49. Improve PHP session cookie security. (2016, Abril 21). Retrieved from <https://www.simonholywell.com/post/2013/05/improve-php-session-cookie-security/>
50. Google reCAPTCHA. (2016, Maio 12). Retrieved from <https://www.google.com/recaptcha/intro/index.html>
51. 10 Usability Heuristics for User Interface Design. (2016, Maio 23). Retrieved from <https://www.nngroup.com/articles/ten-usability-heuristics/>
52. Breadcrumbs In Web Design: Examples And Best Practices. (2016, Maio 23). Retrieved from <https://www.smashingmagazine.com/2009/03/breadcrumbs-in-web-design-examples-and-best-practices/>