



UNIVERSIDADE
LUSÓFONA

GoWake

Trabalho Final de Curso

Relatório Intercalar 2º Semestre

DEISI297

Nome do Aluno: Joel Bernardino (a21806180)

Nome do Aluno: Gabriel Souza (a21800463)

Nome do Orientador: Pedro Serra

Trabalho Final de Curso | LEI | 30 Jun 2023

Direitos de cópia

GoWake, Copyright de (Joel Bernardino e Gabriel Souza), UL.

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

Resumo

Este trabalho arquiteta um sistema mais sofisticado e apropriado, uma aplicação, com intuito de auxiliar as competições de Wakeboard, diferenciando os atletas através de baterias (heats), cujo principal objetivo é ter uma atualização, em tempo real, das pontuações dos respetivos atletas com intuito do mesmo ser usado, futuramente, pela International Waterski & Wakeboard Federation (IWWF).

Os atletas que participam numa competição são distribuídos por categorias (idades), num sistema de baterias (grupo de atletas) e rondas. Os mais pontuados de cada bateria, passam à ronda seguinte até chegar à final.

Esta aplicação web faz a gestão de todas as competições da modalidade Wakeboard, gestão essa composta por:

- Criar evento
- Associar atletas e oficiais (júri) ao evento
- Distribuir atletas por categorias
- Criar rondas e distribuir atletas por baterias
- Associar oficiais às baterias
- Registrar pontuações atribuída pelos jurados a cada atleta para cada bateria
- Gestão da passagem às rondas seguintes

Pretende-se que este trabalho seja colocado em produção e possa ser utilizado pela Federação Internacional de Waterski e Wakeboard.

Devido à complexidade e importância do trabalho, decidiu-se dividir o projeto em duas componentes principais: backend e frontend. Este trabalho foca-se sobretudo no frontend, enquanto o backend foi desenvolvido no âmbito do DEISI 314.

Abstract

This project aims to architect a more sophisticated and appropriate system, an application, to assist Wakeboard competitions by differentiating athletes through heats (batteries), with the main objective of providing real-time updates on the respective athletes' scores. The intention is for this application to be used in the future by the International Waterski & Wakeboard Federation (IWWF).

The athletes participating in a competition are divided into categories (age groups), within a system of heats (groups of athletes) and rounds. The highest-scoring athletes from each heat advance to the next round until reaching the final.

This web application manages all Wakeboard competitions and includes the following features:

- Create events.
- Associate athletes and officials (judges) with the event.
- Distribute athletes into categories.
- Create rounds and assign athletes to heats.
- Associate officials with heats.
- Record scores given by judges to each athlete for each heat.
- Manage progression to the next rounds.

The goal is to deploy this work into production and make it available for use by the International Waterski & Wakeboard Federation. Due to the complexity and significance of the project, it was decided to divide it into two main components: backend and frontend. This work focuses mainly on the frontend, while the backend has been developed as part of the DEISI 314.

Índice

Resumo.....	iii
Abstract	iv
Índice.....	v
Lista de Figuras.....	vii
Lista de Tabelas	viii
1 Identificação do Problema	9
1.1 Visão.....	9
1.2 Objectivos.....	9
1.2.1 Objectivos Gerais:	9
1.2.2 Objectivos Tecnológicos.....	10
1.3 Alcance	10
1.4 Organização.....	10
2 Viabilidade e Pertinência.....	12
2.1 Viabilidade.....	12
2.2 Pertinência	12
3 Benchmarking.....	15
4 Engenharia.....	17
4.1 Levantamento e análise dos Requisitos	17
4.1.1 Descrição dos requisitos de negócio	17
4.1.2 Requisitos Funcionais	19
4.1.3 Requisitos Não Funcionais	23
4.2 Diagrama de Caso de Uso.....	24
4.3 Diagrama de Actividade	25
4.4 Modelos relevantes.....	25
4.5 Estrutura.....	25
5 Solução Proposta.....	26
5.1 Introdução.....	26
5.2 Descrição do funcionamento	26
5.3 Arquitetura.....	28
5.4 Tecnologias e Ferramentas Utilizadas.....	29

5.5	Implementação	31
5.6	Abrangência	33
6	Método e Planeamento	34
7	Resultados	35
8	Conclusão e trabalhos futuros	40
8.1	Bibliografia	40
8.2	Anexos	40
8.2.1	Questionário.....	40
8.2.2	Progresso do trabalho	41
8.3	Glossário.....	42

Lista de Figuras

Figura 1 - A hierarquia por setor do projeto	11
Figura 2 – Utilizadores	13
Figura 3 - Nível de satisfação dos utilizadores	13
Figura 4 - Nível de satisfação dos utilizadores (2)	13
Figura 5 - Utilizador tester 2	14
Figura 6 - Utilizador tester 1	14
Figura 7 - Utilizador tester 3	14
Figura 8 - Diagrama de Casos de Uso	24
Figura 9 - Diagrama BPMN	25
Figura 10 - Mapa aplicacional	25
Figura 11 - Sistema de fusão	27
Figura 12 - Tecnologias e arquitetura adotada no projeto	28
Figura 13 - Crescimento do uso do React	29
Figura 14 - Exemplo conexão API	31
Figura 15 - Endpoints API	32
Figura 16 - Response 200	32
Figura 17 - Response 400	32
Figura 18 - Planeamento do projeto	34
Figura 19 - Cronograma em formato Gantt	34

Lista de Tabelas

Tabela 1 - Resposta ao questionário de melhoria	14
Tabela 2 - Plataformas	16
Tabela 3 - Navegadores compatíveis	33
Tabela 4 - Casos de testes	35
Tabela 5 - Utilizadores	40
Tabela 6 - Grau de usabilidade	40

1 Identificação do Problema

As aplicações atualmente utilizadas pela IWWF apresentam algumas limitações, algumas encontram-se desatualizadas e outras possuem uma interface pouco intuitiva para os utilizadores. Estas aplicações existentes adotam uma abordagem genérica no que diz respeito aos desportos aquáticos, ou seja, não estão direcionadas para uma modalidade específica. O facto de serem generalizadas impede que satisfaçam todos os requisitos específicos e não oferecem o melhor design ou interface para a modalidade em questão, neste caso, o Wakeboard.

Razão pela qual este projeto visa mitigar esses problemas através do uso de linguagens e frameworks modernos, com o objetivo de fornecer o melhor conteúdo e interface para a comunidade de Wakeboard.

Comparativamente à proposta inicial, devido a restrições de tempo e a alguns desafios técnicos encontrados durante o desenvolvimento do projeto, surgiram algumas diferenças e foram realizados ajustes e alterações na proposta original.

Inicialmente, tinha sido estabelecida a pré-condição de que um utilizador necessitaria de possuir um código de oficial da IWWF para efetuar o registo. No entanto, essa exigência foi revogada, uma vez que tornaria a experiência dos utilizadores mais complicada e impediria o acesso ao “Live Results”.

Isso deve-se ao facto de que o “endpoint” (local onde há interação de um serviço) necessitar do ID da competição, sendo impossível obtê-lo sem efetuar o login. Assim, optou-se por remover o “Live Results” da página inicial, sendo agora denominado por “LeaderBoard”.

Devido a restrições de tempo, não foi possível desenvolver uma interface que permitisse a atribuição de permissões aos restantes utilizadores. Contudo, essa tarefa poderá ser realizada através do servidor.

Estas modificações foram ponderadas com base em critérios como usabilidade, funcionalidade, desempenho e cumprimento dos requisitos específicos do Wakeboard, com o intuito de proporcionar a melhor experiência possível aos utilizadores.

1.1 Visão

Uma aplicação web que permite a gestão e a monitorização de competições de Wakeboard.

1.2 Objectivos

1.2.1 Objectivos Gerais:

- Ser integrada em outras aplicações.
- A aplicação fornece uma interface para o utilizador, capaz de funcionar em diferentes tipos e tamanhos de écrans.
- A aplicação automatiza completamente os processos de distribuição de atletas por categoria, determinação do número de baterias por ronda e distribuição de atletas por

bateria. Também automatiza os processos de cálculo de resultados e determinação da eliminação ou avanço de atletas para as rondas seguintes.

- Permitir, aos vários elementos que compõem o júri de uma prova, a introdução das pontuações de cada atleta, em tempo real.
- Gerar relatórios de resultados de cada ronda em tempo real.

1.2.2 Objectivos Tecnológicos

- A aplicação foi desenvolvida usando um pipeline de Continuous Delivery / Continuous Integration [CICD22]
- A aplicação foi desenvolvida em 2 módulos:
 - Backend (API REST) – desenvolvida usando Django Rest Framework
 - Frontend (User Interface) – desenvolvida usando React

1.3 Alcance

Com este projeto pretende-se alcançar a desativação completa do software Waterski Scoring Full [WSFL22] em todas as competições oficiais da modalidade de Wakeboard.

1.4 Organização

A organização do projeto é constituída na primeira camada pelo patrocinador, IWWF, representado por Nuno Eça, presidente do comité mundial de Wakeboard da IWWF.

A segunda camada é composta pelo revisor do projeto, Alexey Philippov, que tem como função rever todo o trabalho feito pelas camadas abaixo até o produto final ser entregue ao patrocinador do projeto.

Em seguida, na terceira camada, temos o orientador, professor Pedro Serra, que é responsável pela gestão e verifica se os requisitos propostos estão a ser cumpridos e respeitados por ambas as equipas, front-end (DEISI 297) e back-end (DEISI 314).

A equipa de front-end é composta pelos alunos e desenvolvedores Joel Bernardino e Gabriel Souza que são responsáveis pela gestão de uma UI (User Interface) de alta performance aos utilizadores finais.

Seguidamente, a equipa de back-end é constituída pelos alunos e desenvolvedores Miguel Soares e Mauro Monteiro que estão responsáveis pela criação de uma API REST [APRT22] que faz a gestão dos dados envolvidos na aplicação.

A Figura 1 demonstra a hierarquia do projeto proposto pelo IWWF.

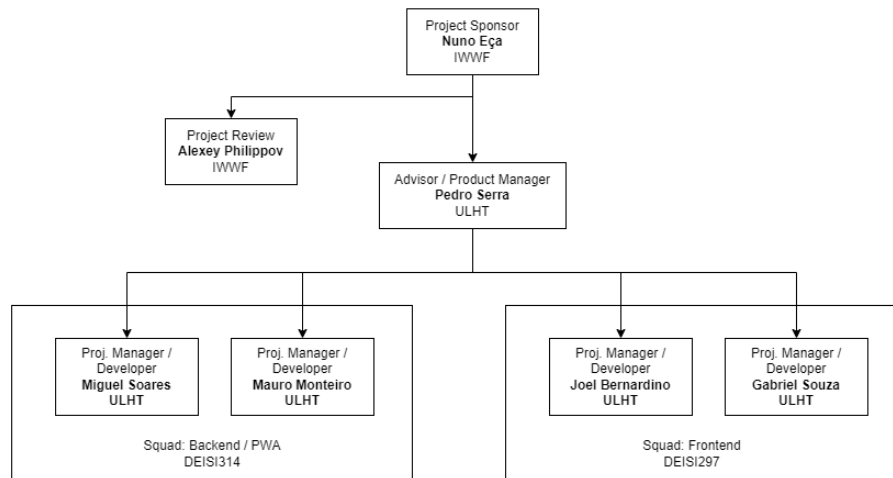


Figura 1 - A hierarquia por setor do projeto

2 Viabilidade e Pertinência

2.1 Viabilidade

A aplicação pretende colmatar os problemas identificados e será colocada em produção para ser utilizada pela federação portuguesa de Wakeboard.

Este projeto não possui qualquer custo e visa proporcionar aos utilizadores um modelo e funcionalidades que satisfaçam e melhorem os requisitos específicos, atendendo às necessidades da modalidade Wakeboard.

Para isso, são aplicadas as melhores práticas de UX (Experiência do Utilizador) e usabilidade, com foco na responsividade, garantindo uma experiência aprimorada para o utilizador em qualquer dispositivo eletrónico utilizado.

Atualmente, a aplicação, já se encontra disponível no repositório do DEISI e no servidor organizacional da IWWF.

Caso os resultados sejam positivos, a aplicação poderá ser expandida e distribuída como uma aplicação de código aberto (open source) ou para uso internacional.

2.2 Pertinência

Para comprovar a relevância do projeto no contexto anteriormente descrito, foi realizado um questionário aos utilizadores do GoWake. Isso possibilitou identificar de forma clara e objetiva quais são os principais problemas que estão a afetar o desempenho, permitindo, assim, adotar medidas mais assertivas para a sua melhoria.

O objetivo deste questionário é medir os níveis de satisfação e as dificuldades dos utilizadores, com base nos princípios gerais de usabilidade. Foi avaliada a navegabilidade, consistência, clareza visual e envolvimento dos utilizadores com a aplicação.

Além disso, procurou-se identificar eventuais problemas de utilização e erros, tanto através da observação direta como da análise posterior dos dados reportados pelos utilizadores.

É importante salientar que os utilizadores testaram a aplicação de duas maneiras distintas:

- Conta pública: Os utilizadores registaram-se e acederam ao site utilizando as suas respetivas contas, sem possuírem qualquer tipo de permissões para efetuar alterações ou gerir eventos.
- Conta de “Calculator”: Foi fornecido aos utilizadores uma conta com permissões adicionais, possibilitando-lhes testar a aplicação na sua totalidade.

É de realçar que todos os testes foram realizados presencialmente, no “localhost”, utilizando o portátil de um dos membros do projeto. As respostas foram registadas e serão mostradas nas seguintes figuras:

A Figura 2 apresenta os utilizadores que testaram a plataforma GoWake, sendo seis do sexo masculino e três do sexo feminino, totalizando nove utilizadores.

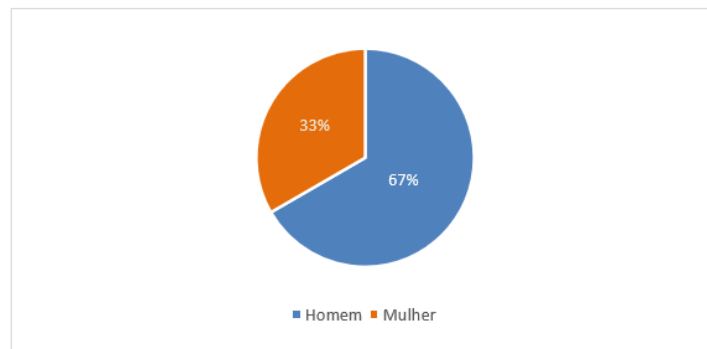


Figura 2 – Utilizadores

A Figura 3 e a Figura 4 apresentam os resultados do questionário de satisfação, fornecendo informações valiosas sobre a opinião dos utilizadores, assim como dados objetivos sobre o seu comportamento e preferências.

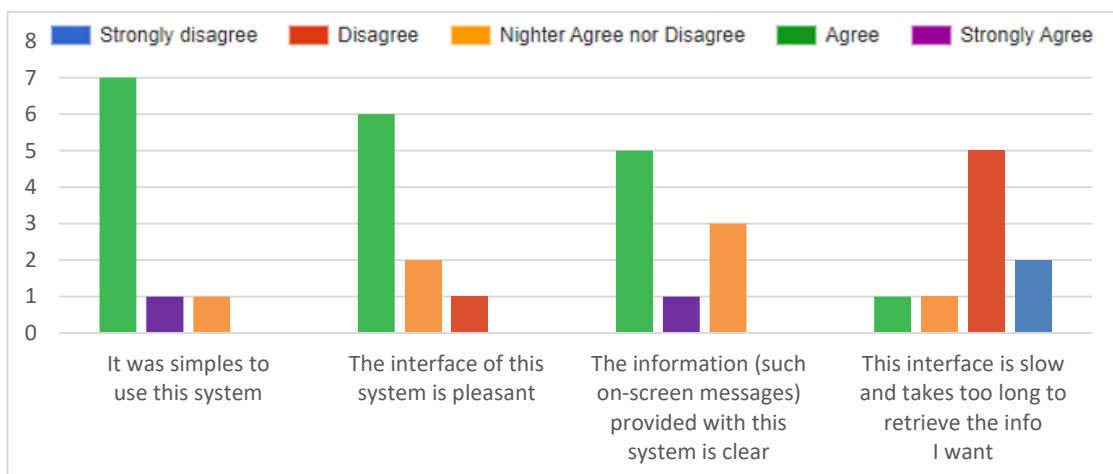


Figura 3 - Nível de satisfação dos utilizadores

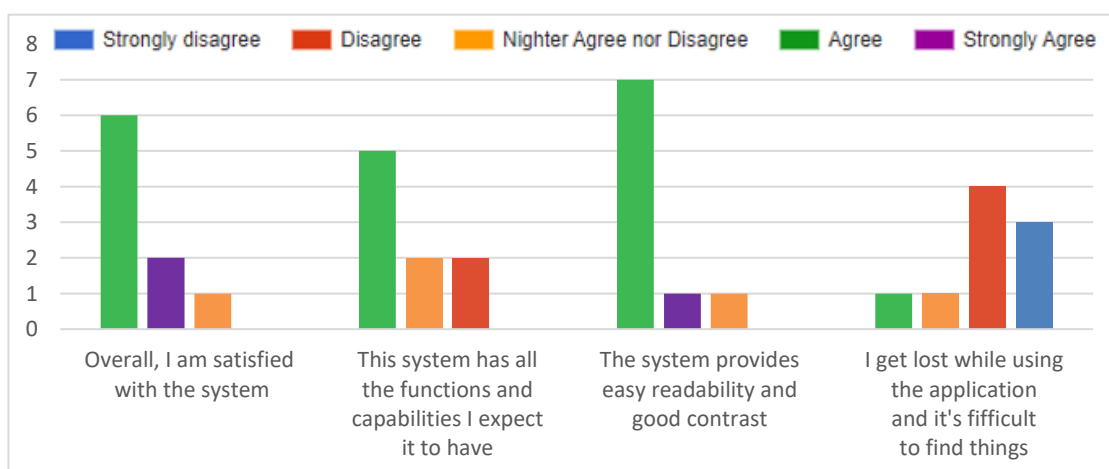


Figura 4 - Nível de satisfação dos utilizadores (2)

A Tabela 1 mostra as respostas aos utilizadores face as sugestões de melhoria na aplicação.

Tabela 1 - Resposta ao questionário de melhoria

Indique sugestões de melhoria sobre o atual funcionamento da aplicação
As mensagens de erro ao carregar o ficheiro XML não são claras
Podia haver uma página ou algum tipo de imagem que tivesse contacto ou email de alguém que nós pudessemos recorrer em caso de algum constrangimento ou erro
Ter uma página de perfil acho que é sempre bom
A mensagem quando se faz login com sucesso demora muito a desaparecer
A página do XML fica deformada se mudar a dimensão do ecrã
Mais animação para ser mais interativo
Podiam destacar mais a modalidade que está a ser avaliada

É de realçar que o questionário e a questão de resposta aberta foram realizados momentos antes do término do projeto, dando-nos uma pequena margem para implementar melhorias com base no feedback recebido dos utilizadores.

Segue, em seguida, as imagens de alguns dos participantes que testaram a aplicação e responderam ao inquérito e a pergunta de resposta aberta.

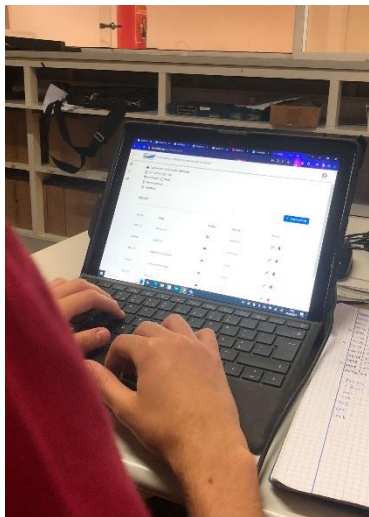


Figura 6 - Utilizador tester 1

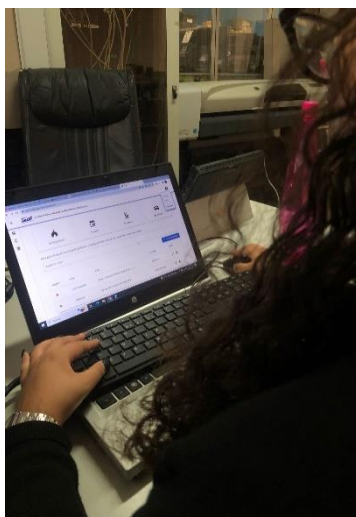


Figura 5 - Utilizador tester 2

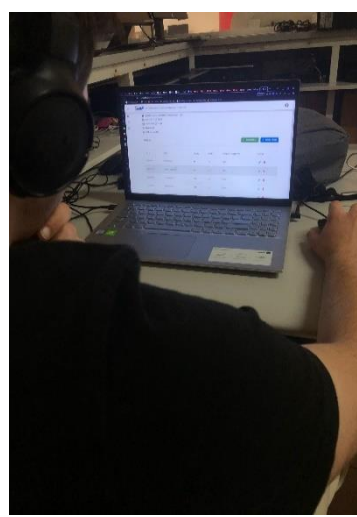


Figura 7 - Utilizador tester 3

3 Benchmarking

No mercado atual existem poucas plataformas de software para gestão de competição de desportos radicais aquáticos, porém elas carecem de um modelo e qualificações que possam satisfazer e melhorar os requisitos específicos para a modalidade Wakeboard.

Por não apresentarem requisitos específicos para a respectiva modalidade fazem com que as mesmas sejam menos eficientes, resultando de falta de informação, múltiplos cliques desnecessários e uma interface menos “user friendly”. Em consequência, por não assegurarem que o regulamento é cumprido, não permitem a homologação das provas.

Exemplo das plataformas, outrora acima citadas, são:

Liveheats - oferece transmissões e resultados ao vivo de diversas modalidades, como surf, cable-ski, skate, entre outros. O facto do mesmo ser abrangente, ou seja, generalizada, não disponibiliza todos os requisitos específicos para uma única modalidade, wakeboard. Requisitos esses que contribuem para uma melhor interface do utilizador.

Dito isto, a aplicação possui algumas desvantagens no qual o propósito deste projeto visa a mitigar e solucionar os mesmos:

- **Custo** - o Liveheats é uma solução paga, e o custo pode variar dependendo do volume de dados e recursos necessários. Para pequenas empresas ou empreendedores individuais com orçamentos limitados, isso pode ser uma desvantagem significativa.
- **Curva de aprendizagem** - embora a interface do Liveheats seja fácil de usar, pode levar algum tempo para os utilizadores se acostumarem com a plataforma e entender todas as suas funcionalidades e recursos.
- **Navegação complexa** - para utilizadores iniciantes, a navegação na interface do Liveheats pode ser complexa e confusa, pois há muitas opções e menus diferentes para explorar.
- **Visualização de dados** - embora o Liveheats ofereça muitas informações úteis e percepções valiosas, a visualização dos dados pode ser desafiadora para alguns utilizadores. Dependendo do tipo de gráfico ou tabela usada para apresentar as informações, pode ser difícil entender as tendências e padrões de forma clara e concisa.

Waterski Scoring 2022 - A aplicação permite que os jurados insiram pontuações para o desempenho de cada atleta, com base em vários fatores, como a distância percorrida, a velocidade, a técnica e a dificuldade da manobra realizada.

Ou seja, ajudou a melhorar a qualidade e a precisão dos resultados em competições de wakeboard, eliminando a necessidade de cálculos manuais e minimizando erros. Disponibiliza outras funcionalidades úteis, como a criação de eventos a partir de um ficheiro XML e a gestão de heats (baterias).

Em semelhança à ferramenta líder, LiveHeats, esta aplicação, Waterski Scoring 2022, carece de uma navegabilidade mais fácil, simples e apelativa e de uma interface mais “user-friendly”.

No entanto é importante salientar que o projeto foi desenvolvido exclusivamente para atender as necessidades da modalidade Wakeboard, tanto em questões de segurança, usabilidade, flexibilidade e interoperabilidade.

A Tabela 2 sumariza as diferenças entre duas plataformas existentes, sendo uma delas a líder atual (liveheats), a outra que está atualmente a ser usada nas competições Wakeboard Portugal (Waterski Scoring 2022) e a que foi implementada neste projeto (GoWake).

Tabela 2 - Plataformas

Plataformas			
Classificadores	GoWake	Waterski Scoring 2022	LiveHeats
Plano pagamento	Gratuito	Gratuito	Varia consoante o tipo de evento
Valor do evento principiante (até 50 atletas)	Gratuito	Gratuito	600€
Valor do evento intermédio (até 100 atletas)	Gratuito	Gratuito	1000€
Valor do evento profissional (até 200 atletas)	Gratuito	Gratuito	1600€
Valor do evento elite (até 500 atletas)	Gratuito	Gratuito	2500€
Recebe XML	Sim	Sim	Não
User friendly	Sim	Não	Sim
Acessibilidade	Sim	Sim	Não
Disponibilidade	Sim	Não	Não

4 Engenharia

A estrutura é a organização ou arquitetura interna de um sistema, produto ou serviço sendo possível incluir componentes, subsistemas e relações entre eles. A escolha da estrutura é influenciada pelos requisitos, pois ela é capaz de atender as necessidades e proporcionar uma boa performance.

4.1 Levantamento e análise dos Requisitos

Os requisitos são as necessidades ou especificações que um sistema, produto ou serviço precisa atender para ser considerado válido e útil.

4.1.1 Descrição dos requisitos de negócio

Os requisitos de negócio são as necessidades e expectativas que um negócio precisa atender para alcançar seus objetivos e ser bem-sucedido, dito isto, foi anexado uma lista de requisitos disponibilizada pela própria IWWF razão pela qual optamos por citá-los na sua forma original.

RN001 - The application shall be web-based, accessible via browser on a desktop computer or via mobile browser.

RN002 - The application shall have a public interface, accessible to anyone, and private interface accessible only to registered officials with a valid username/password combination.

RN003 - The app shall have 5 levels of permission:

- 1 – public (without password): can access information such as ongoing and past events, results, participating athletes, etc.
- 2 – Official (jury) (password required): can view and modify scores of the ongoing heats in which access was granted to
- 3 – Calculator (password required): can create events and manage events where he/she was added with calculator rights.
- 4 – Chief judge (password required): can create events and manage events, assigns calculators to events. Especially for overseeing purposes of the running of the event. Can escalate permissions of other users up to calculator.
- 5 – Administrator (password required): can manage all events, can add escalate power of calculators within events.

RN004 - Any user with calculator or chief judge rights can create an event by importing the event file generated by the Event Management System (EMS).

RN005 - A user above level 3 rights within a created event can add and manage officials and athletes.

RN006 - Each event has an overview page, available to the public, which includes, at least: a picture, site description, dates, live scores, results, rounds, participating athletes, divided into categories, rounds and heats.

RN007 - After the automatic event creation (from the EMS) which shall include all the athletes and categories, the app shall generate the start list of the first round and of all the subsequent rounds until the end of the competition, according to the IWWF rules.

RN008 - The app shall automatically create a timetable for the different rounds but shall allow changes to be applied by any user with calculator rights.

RN009 - The app shall allow the calculator to assign the official panel to each round or heat.

RN010 - The app shall provide a private interface for each jury to introduce his scores according to the format specified in the IWWF rules: Execution, Composition, Intensity, in the case of a boat wakeboard competition. Different formats will be included for Cable wakeboard and Wakesurf as needed.

RN011 - The calculator user can also introduce the scores or override the scores introduced by each individual jury.

RN012 – A Judge can only introduce scores in heats where he/she was assigned to by the calculator.

RN013 - The app shall provide an export to xml interface that exports the event results in a format possible to be imported by the EMS.

RN014 - A dedicated mobile front end shall be available for the jury to introduce scores and compare results.

RN015 - The app shall be developed with a clear separation between back end and front-end, where the backend is fully accessible via RESTfull interface.

RN016 - The app shall be able to export the start lists for each round in a printable pdf format.

RN017 - Source code rights over the developed code are owned by the IWWF. IWWF if free to decide to make the source-code open under any license.

RN018 – The app provides a view specially designed for a dock TV with reels of starting lists, results, heat systems, etc.

4.1.2 Requisitos Funcionais

Os requisitos funcionais são utilizados para definir as necessidades do utilizador e como o sistema, produto ou serviço deve comportar-se. São fundamentais para garantir que o sistema, produto ou serviço atenda às necessidades do utilizador e seja útil e eficaz.

RF001 – Login de utilizador na aplicação web. (Must have) (Implementado)

Descrição: Deverá ser disponibilizada uma opção login de utilizador, para aceder à aplicação.

Pessoa(s): 1 pessoa

Dia(s): 2 dias

Relação com RFs:

- DEISI314_RF.005 – Criação de API endpoints para front end

Precondições: O utilizador deve estar registado.

Resultado esperado: Após a respetiva validação de autenticação do utilizador, o mesmo acede à aplicação com as funcionalidades correspondentes ao seu perfil.

Processo: O Back end deverá garantir a validação da autenticação do utilizador e, caso a mesma se comprove, deve permitir o acesso à aplicação.

RF002 - Registo de novo utilizador na aplicação web. (Must have) (Implementado)(Alterado)

Descrição: Deverá ser disponibilizada uma opção para criação de registo de utilizador, para aceder à aplicação.

Pessoa(s): 1 pessoa

Dia(s): 3 dias

Relação com RFs:

- DEISI314_RF.005 – Criação de API endpoints para front end

Precondições: O utilizador deve ser membro da IWWF, elegível para perfil de Official, Calculator, Administrator.

Resultado esperado: Após validação de autenticação do utilizador, o registo é criado e é concedido acesso à aplicação com as funcionalidades correspondentes ao seu perfil.

Processo: O back end deverá garantir a validação da autenticação do utilizador e criação de novos registos, mediante combinação dos critérios definidos, autenticação essa que inclui a validação de um IWWF_ID (código de oficial da IWWF). Caso o utilizador queira registar-se não tendo essa permissão de acesso, será apresentada uma mensagem de erro a informar que o mesmo não tem permissões para efetuar o registo.

Alteração: foi removido a parte da validação do código de oficial da IWWF pelos motivos já apresentados acima.

RF003 – Recuperação de senha (Must have) (Implementado)

Descrição: Deverá ser disponibilizada uma opção de recuperação de password ao utilizador.

Pessoa(s): 1 pessoa

Dia(s): 2 dias

Relação com RFs:

- DEISI314_RF.005 – Criação de API endpoints para front end

Precondições: O utilizador deve estar registado.

Resultado esperado: Quando solicitada a funcionalidade, é acionado o envio de um email com as instruções para modificação de password.

Processo: O Back end deverá garantir a validação do email do utilizador, e acionar o envio de uma mensagem para o seu respetivo endereço de email, com instruções para alterar a password que, posteriormente, será guardada em substituição da anterior.

RF004 – Carregar ficheiro XML (Must have) (Implementado)

Descrição: Qualquer utilizador com permissões de calculator pode criar um evento fazendo o upload de um ficheiro XML que contém toda a informação da competição.

Pessoa(s): 2 pessoas

Dia(s): 7 dias

Relação com RFs:

- DEISI297_RF.005? – Criação de API endpoints para front end
- DEISI314_RF.006? – Distribuição de Heats do Evento

Precondições: O utilizador deve ser membro da IWWF, elegível para perfil de Calculator.

Resultado esperado: A distribuição automática dos atletas no respetivo evento, será colocada à consideração do Calculator, para que seja modificada se necessário, e, posteriormente, validada.

Processo: O calculator tem à sua mercê, a posteriori, a possibilidade de modificar, de forma manual, retirar e / ou adicionar atletas e oficiais (júri) no evento ao qual ele foi consagrado, no qual, posteriormente, será guardado e validado no Back End.

RF005 – Interfaces create e modify events (Must have) (Implementado)

Descrição: O utilizador com privilégio de calculator tem em sua posse a gestão de eventos.

Pessoa(s): 2 pessoas

Dia(s): 14 dias

Relação com RFs:

- DEISI314_RF.005 – Criação de API endpoints para front end
- DEISI314_RF.006 – Distribuição de Heats do Evento

- DEISI314_RF.007 – Adicionar e gerir oficiais e atletas

Perfil de utilizador: Calculator

Precondições: O utilizador dever ser membro da IWWF, com perfil Calculator.

Resultado esperado: Os oficiais e atletas adicionados e / ou modificados serão enviados a API e, posteriormente, gravados na base de dados, no qual ficarão disponíveis para serem integrados nos respetivos eventos.

Processo:

1. Modificar atletas e oficiais (júri) em heats – O sistema permitirá o calculator alocar os oficiais para cada ronda ou heat (bateria) que, posteriormente, será guardado e validado pelo sistema.
2. Atribuir pontuação – Há a possibilidade de introduzir pontuações ou substituir as pontuações, outrora, introduzidas pelos jurados.
3. Gerir heats – Deverá ser disponibilizada uma opção que irá adicionar, de forma automática, os heats.

RF006 – Live Results (Must have) (Implementado)(Alterado)

Descrição: Deverá ser disponibilizada uma lista das competições de wakeboard, já finalizadas e/ou a decorrer.

Pessoa(s): 2 pessoas

Dia(s): 10 dias

Relação com RFs:

- DEISI314_RF.005 – Criação de API endpoints para front end
- DEISI314_RF.006 – Distribuição de Heats do Evento

Perfil de utilizador: Publico

Precondições: O utilizador não precisa ter um registo para ter acesso.

Resultado esperado: Visualizar a classificação de distintos heats.

Processo: O utilizador terá a possibilidade de visualizar com mais detalhe as seguintes informações:

1. Os nomes dos respetivos atletas
2. As classificações dos mesmos
3. Os países dos atletas envolvidos na respetiva competição
4. Os pontos totais e os pontos adquiridos em cada heat
5. A pontuação de cada jurado atribuída aos atletas
6. Nome do campeonato / circuito
7. A idade e a categoria do atleta
8. A informação do júri outorgado na respetiva competição
9. Data do evento / competição
10. Estado do evento (por exemplo: a decorrer ou finalizado).

- **Alteração:** agora é intitulado por “Leaderboard” e tem as seguintes funcionalidades:
 1. **Listar todos os atletas agrupados pelas categorias do respetivo evento**
 2. **Listar a pontuação de cada atleta**
 3. **Possibilidade de alterar as pontuações**

As restantes informações como os países, idade, data e hora do evento que, deveriam estar contidas no “LiveResult”, estão impressas noutras interfaces.

É necessário o utilizador ter registo e estar ‘logado’ para ter acesso ao mesmo.

RF007 – Recursos (Must have) (Implementado) (Alterado)

Descrição: O utilizador terá ao seu dispor as opções de consultar as regras do wakeboard e as categorias.

Pessoa(s): 1 pessoa

Dia(s): 2 dias

Relação com RFs: N/A

Perfil de utilizador: Publico

Precondições: O utilizador não precisa ter um registo para ter acesso.

Resultado esperado: Visualizar de forma clara informações inerentes as competições de wakeboard.

Processo:

1. **Regras do Wakeboard:** Caso o utilizador pretenda averiguar as regras da respetiva modalidade, terá acesso a um documento PDF incorporado pela IWWF cujo mesmo será aberto em um novo separador web.
2. **Categorias:** O utilizador terá, em sua disponibilidade, o acesso a uma tabela informativa que exibirá as várias categorias existentes na respetiva modalidade.

Alteração: Foi removida a interface das categorias, uma vez que essa informação é redundante e a mesma já se encontra presente no PDF que contém as regras do Wakeboard.

RF008 – Interface Atribuir Permissões (Must have) (Não Implementado)

Descrição: Deverá ser disponibilizada uma opção para atribuir cargos / permissões de Calculator e / ou de oficial.

Pessoa(s): 1 pessoa

Dia(s): 2 dias

Relação com RFs:

- DEISI314_RF.005 – Criação de API endpoints para front end
- DEISI314_RF.006 – Distribuição de Heats do Evento
- DEISI314_RF.007 – Adicionar e gerir oficiais e atletas

Perfil de utilizador: Chief judge

Precondições: O utilizador deve ser membro da IWWF, com perfil Chief judge.

Resultado esperado: O utilizador categorizado como Chief judge deve ter a possibilidade de atribuir e modificar permissões de um utilizador intitulado como calculator e / ou oficial.

Processo: O Chief judge tem à sua mercê, a possibilidade de modificar e atribuir permissões aos vários tipos de eventos.

Não foi implementado devido à restrição de tempo.

4.1.3 Requisitos Não Funcionais

Os requisitos não funcionais são as necessidades ou especificações que um sistema, produto ou serviço precisa atender para ser considerado de qualidade e seguro. Eles descrevem como o sistema, produto ou serviço deve se comportar, sem se preocupar com o que ele deve fazer.

RNF01 – Navegabilidade (Must have)

Descrição: A aplicação web precisa ter uma interface intuitiva e de fácil navegação no qual as informações devem ser encontradas sem grande esforço e de fácil visualização.

Pessoa(s): 1 pessoa

Dia(s): 2 dias

RNF02 – Adaptabilidade (Must have)

Descrição: Garante se o sistema consegue identificar novos documentos, nomeadamente mediante a capacidade para poder considerar novos modelos de documentos.

Pessoa(s): 2 pessoas

Dia(s): 4 dias

RNF03 – Segurança (Must have)

Descrição: Garante as condições que o sistema deve garantir nomeadamente a proteção de dados.

Pessoa(s): 2 pessoas

Dia(s): 6 dias

RNF04 – Conectividade (Must have)

Descrição: Para aceder as informações atualizadas a aplicação web deverá ter conexão a internet.

Pessoa(s): 1 pessoa

Dia(s): 3 dias

4.2 Diagrama de Caso de Uso

A Figura 8 mostra um diagrama de caso de uso, que documenta o que o sistema faz do ponto de vista do utilizador. Em outras palavras, descreve as principais funções do sistema e a interação das mesmas com os utilizadores.

O Diagrama é composto basicamente por quatro partes:

- Cenário: Sequência de eventos que acontecem quando um utilizador interage com o sistema.
- Ator: Utilizador do sistema, ou melhor, um tipo de utilizador.
- Caso de Uso: É uma tarefa ou uma funcionalidade realizada pelo ator (utilizador).
- Comunicação: É o que liga um ator com um caso de uso.

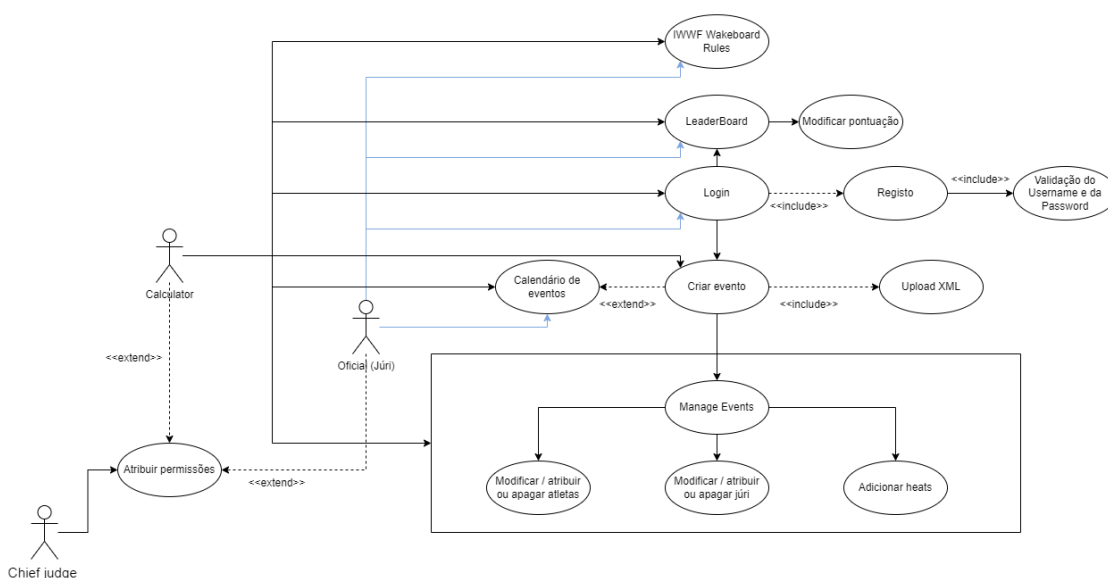


Figura 8 - Diagrama de Casos de Uso

- O chief judge tem todos os privilégios do sistema sendo responsável por atribuir permissões de calculator e oficial e, além disso, ele tem a autoridade de reverter decisões de qualquer tipo de utilizador.
- O calculator pode criar um evento e atribuir e / ou modificar tanto oficiais como atletas para uma determinada competição, além do mais, tem em sua mercê o poder de reavaliar as pontuações que, outrora, foram atribuídas pelos oficiais podendo fazer tudo o que um oficial pode.
- O oficial tem apenas permissão para pontuar os atletas de uma determinada competição, ao evento do qual o mesmo foi atribuído pelo calculator, através da aplicação móvel desenvolvida pelo DEISI314 com o acréscimo do mesmo poder aceder as funcionalidades de consultas.

4.3 Diagrama de Actividade

Usamos o BPMN (Business Process Model and Notation) pois é uma notação gráfica para modelar processos de negócios, que fornece uma forma clara e compreensível de documentar e comunicar fluxos de trabalho de negócios como demonstra a Figura 9.

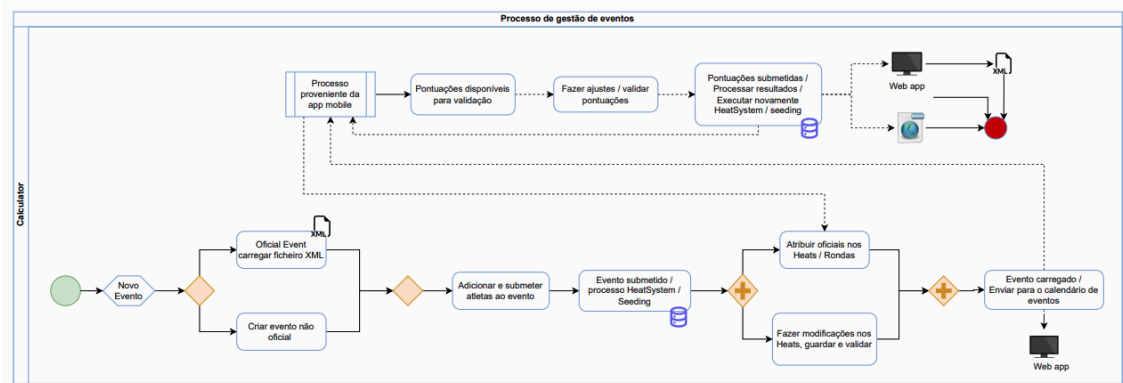


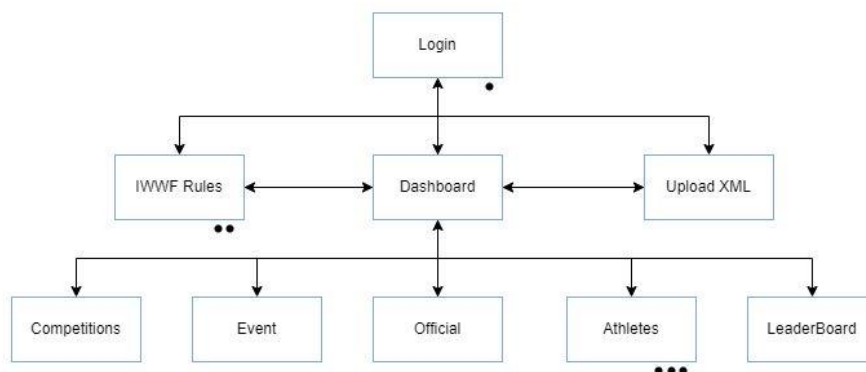
Figura 9 - Diagrama BPMN

4.4 Modelos relevantes

O diagrama de Entidade-Relação e o Modelo de Classes foi desenvolvido pelo back-end (DEIS1314). A união de todos esses modelos é a arquitetura completa da aplicação que nos foi proposta para o desenvolvimento.

4.5 Estrutura

A Figura 10 é um mapa aplicativo que é uma representação visual da estrutura da aplicação web que será desenvolvida, mostrando como os diferentes componentes se relacionam e interagem.



- Existe a possibilidade do utilizador criar conta e / ou fazer a recuperação da password
- O utilizador poderá aceder as páginas sem efetuar o login
- Existe a opção do utilizador gerar heats

Figura 10 - Mapa aplicativo

5 Solução Proposta

5.1 Introdução

A solução perante as advertências nas aplicações já existentes foi desenvolver uma aplicação web, exclusivamente para atender as necessidades da modalidade Wakeboard, que fornece informações sobre os atletas e as suas respetivas pontuações em cada ronda, a informação dos jurados e toda a gestão das competições de wakeboard.

O objetivo sempre foi de disponibilizar uma melhor experiência ao utilizador, porém o desempenho do site dependia de vários fatores como a quantidade de dados solicitados, a latência da API, a velocidade da conexão com a API e a eficiência do código implementado no frontend.

A solução foi desenvolver um carregamento assíncrono para buscar dados da API que permitia que outras partes do site continuassem a ser carregadas e exibidas enquanto os dados eram buscados em segundo plano.

O desenvolvimento da mesma foi implementado em React, disponibilizando uma melhor interface, usabilidade, flexibilidade e interoperabilidade.

Em baixo listamos:

- Repositório do github: <https://github.com/DEISI-ULHT-TFC-2022-23/TFC-DEISI297-GoWake-Front-End>
- Link para video: <https://youtu.be/l4bAObt7kzo>
- Link da aplicação em produção: <https://iwwf.netlify.app>

Descreve-se de seguida a estrutura desta secção:

- Na secção 5.2 faz-se uma breve descrição sobre o funcionamento da modalidade.
- Na secção 5.3 é consignado a arquitetura e os componentes usados na implementação do projeto.
- Em seguida, secção 5.4, é relatado as tecnologias e ferramentas usadas no projeto e a importância e o motivo do uso das mesmas.
- Posteriormente a secção 5.5 mostra tudo o que foi implementado.
- Por fim, a secção 5.6, dita o quão impactante e como as disciplinas contribuíram no decorrer deste projeto.

5.2 Descrição do funcionamento

O desenvolvimento da tecnologia está a crescer rapidamente nas últimas décadas e tem um impacto no uso de gadgets e nas atividades diárias. À medida que a tecnologia sofisticada é utilizada, as necessidades humanas também estão a mudar cada vez mais. Muita tecnologia de pesquisa tem sido usada para facilitar e acelerar os trabalhos humanos e o mesmo se aplica ao desporto.

Os atletas que participam numa competição são distribuídos por categorias (idades), e os atletas competem dentro da sua categoria num sistema de baterias (grupo de atletas) e rondas. Os mais pontuados de cada bateria, passam à ronda seguinte até chegar à final.

Haverá um mínimo de 3 atletas necessários para abrir uma categoria. As categorias a serem abertas são determinadas após o término do registo.

Um atleta poderá estar associado a uma única categoria, categoria essa que poderá ser da sua respetiva faixa etária ou de nível de dificuldade superior à sua própria faixa etária.

Cada categoria é definida por:

- U14 – Rapazes / Raparigas – Caso a idade do atleta seja 14 ou inferior até 31 de dezembro do ano em que o evento irá realizar.
- U18 – Júnior Rapazes / Júnior Raparigas - Caso a idade do atleta seja 18 ou inferior até 31 de dezembro do ano em que o evento irá realizar.
- O30 – Mestre Homem / Mestre Mulher - Caso a idade do atleta seja 30 ou superior até 31 de dezembro do ano em que o evento irá realizar.
- O40 – Veterano Homem / Veterano Mulher - Caso a idade do atleta seja 40 ou superior até 31 de dezembro do ano em que o evento irá realizar.
- Divisão Aberta (Homem / Mulher) – Não há restrição de idades.

No caso de uma categoria não abrir, por falta de participantes, o Juiz Chefe informará a todos sobre a respetiva insuficiência e haverá uma fusão. Ou seja, categorias com menos de três atletas serão canceladas ou englobadas para a categoria seguinte / anterior.

A Figura 11 mostra um exemplo de como funciona o sistema de fusão:



Figura 11 - Sistema de fusão

Os jurados anotam todas as habilidades executadas pelo atleta. Ao pontuar, os seguintes fatores também são considerados:

- Grau de dificuldade de cada truque
- Estilo do atleta
- Suavidade de execução
- Desempenho geral do atleta

Existe um conjunto de regras muito específicas definidas no regulamento da federação. No qual um evento só pode ser homologado (gerar pontos para o ranking oficial dos atletas) caso as regras do regulamento tenham sido cumpridas. O software tem um papel fulcral pois implementa as regras definidas no regulamento, garantido assim, que os eventos que tenham sido realizados recorrendo ao software aqui desenvolvido, poderão ser homologados.

5.3 Arquitetura

A arquitetura do projeto é baseada em um modelo cliente-servidor, onde um cliente envia uma solicitação a um servidor e o servidor retorna uma resposta utilizando métodos HTTP como GET, POST, PUT e DELETE para interagir com dados e recursos. Esses métodos são usados para obter, criar, atualizar e excluir dados, respectivamente.

É uma abordagem simples, flexível e fácil de usar e pode ser introduzida praticamente por qualquer linguagem de programação. É muito usado para construir serviços web devido a sua ampla escalabilidade e simplicidade.

A ligação à API (Application programming Interface), cujo intuito é ter uma conexão entre o Front-End e o Back-End, é o REST (Representational State Transfer) que é um estilo de arquitetura para construir serviços web.

A Figura 12 descreve a organização dos componentes, as relações entre eles, as interfaces e a lógica de negócios que gere o projeto no qual a parte de back-end e database (base de dados), é gerida pelo DEISI314.

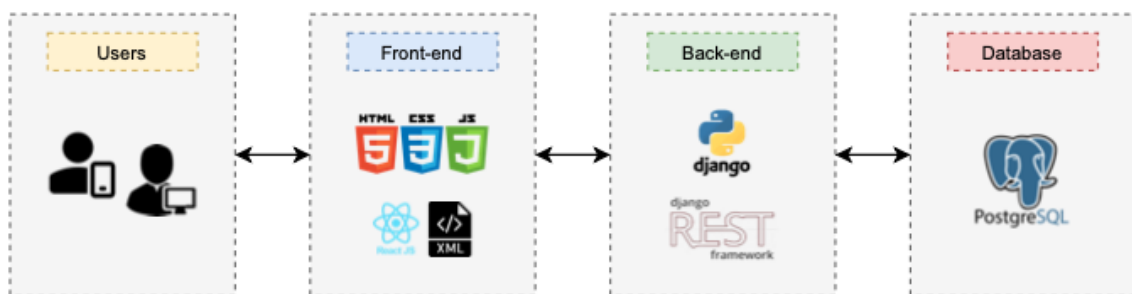


Figura 12 - Tecnologias e arquitetura adotada no projeto

O front-end é responsável por apresentar a interface do utilizador e comunicar-se com o back-end através de requisições HTTP [HTTP22]. O back-end é responsável por processar as requisições, acessar a base de dados e fornecer as respostas apropriadas.

As tecnologias utilizadas no front-end são HTML (Linguagem de Marcação de HiperTexto), CSS (Cascading Style Sheets) e o React que é uma biblioteca JS (JavaScript) de open source (código aberto).

O XML (Extensible Markup Language) é utilizado apenas para realizar o carregamento de dados por parte dos administradores na plataforma para depois ser vinculada ao back-end e em seguida a base de dados.

5.4 Tecnologias e Ferramentas Utilizadas

Sendo esta parte do projeto detentora do front-end, além do HTML e CSS, é utilizado o React [REACT22] que é uma biblioteca JavaScript de código aberto. Por ser mais preciso e flexível, a sua componentização melhora a eficiência nas atualizações de páginas web, razão pela qual optamos pelo seu uso, cujo objetivo é disponibilizar uma aplicação web responsiva e interativa.

Existem diversos motivos que tornam o React Js tão popular entre os desenvolvedores, como:

- A possibilidade de componentizar aplicações.
- A eficiência em gerar as páginas web.
- A forma como reage às mudanças, atualizar e renderizar apenas uma parte da tela em vez da página toda.
- Utilizar uma linguagem declarativa no código.
- Ser uma biblioteca JavaScript, uma das linguagens mais famosas do mundo e mais utilizadas.
- Ser móvel, podendo ser usada também em aplicações Mobile com o React Native, entre outras.

A Figura 13 demonstra o quão bem o React tem crescido consideravelmente entre os desenvolvedores e, conseqüentemente, no mercado, sendo uma das principais escolhas para o desenvolvimento de aplicações web e mobile.

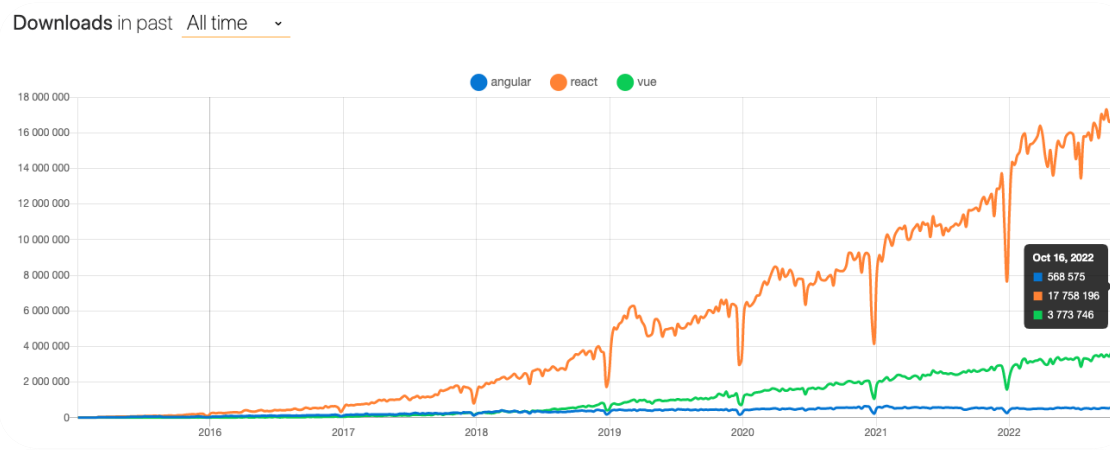


Figura 13 - Crescimento do uso do React

Todas as funcionalidades do React são suportadas pelos navegadores modernos, incluindo:

- **Componentização:** o React permite criar componentes reutilizáveis e organizados de forma hierárquica. Esse recurso é suportado por todos os navegadores modernos.
- **Virtual DOM:** o React utiliza o Virtual DOM para otimizar o desempenho da renderização de componentes. Esse recurso é suportado por todos os navegadores modernos.
- **JSX:** o React permite escrever código HTML diretamente no JavaScript, utilizando a sintaxe JSX. Esse recurso é suportado por todos os navegadores modernos.
- **State e Props:** o React utiliza o conceito de state e props para gerir o estado e as propriedades dos componentes. Esse recurso é suportado por todos os navegadores modernos.
- **React Hooks:** os React Hooks são uma forma de utilizar state e outras funcionalidades do React em componentes funcionais. Esse recurso é suportado pelos navegadores modernos, com exceção do Internet Explorer.

Uma das ferramentas essenciais no decorrer do projeto foi o **Postman** que é uma ferramenta poderosa para o desenvolvimento de API que funciona como uma interface gráfica do utilizador, que suporta operações assíncronas, pois permite criar e enviar várias solicitações HTTP/HTTPS em sequência, com intuito de testar diferentes aspetos da API e verificar se o comportamento da mesma está correto. Foi a partir dessas operações que nos foi possível estabelecer a comunicação entre o Back-End e o Front-End.



Todo o código fonte no decorrer do projeto foi implementado no **Visual Studio Code** e no **IntelliJ** que são ferramentas populares entre a comunidade de desenvolvedores de aplicações React, sendo a primeira mais adequada para desenvolvedores de React e a segunda mais usada para o desenvolvimento de aplicações em Java.



Usamos o **GitHub** como plataforma de hospedagem de todo o código-fonte no qual possibilita a colaboração e a gestão do projeto. Esta ferramenta oferece uma ampla gama de recursos, como sistemas de controle de versão, solicitações de pull, ferramentas de revisão de código e entre outros. Adicionalmente, também fornece recursos de integração com as ferramentas de desenvolvimento, como as que foram acima mencionadas.



5.5 Implementação

Neste projeto, desenvolveu-se uma aplicação web, que consiste essencialmente num site que se comporta como uma aplicação e é executado através de um navegador. Além disso, a aplicação é responsiva, o que significa que o seu layout se adapta automaticamente ao tamanho do ecrã utilizado, proporcionando uma experiência adequada em dispositivos móveis e diferentes resoluções de ecrã.

Relativamente ao Front-End, foram implementadas as seguintes funcionalidades:

- Login / Logout
- Registo
- Recuperação de senha
- Nav sidebar (Navegação lateral)
- IWWF Wakeboard rules
- Dashboard
- Upload XML
- Leaderboard
- Gestão da competição
- Gerir evento
- Gerir atletas
- Gerir jurados

Descreve-se de seguida um exemplo da implementação do componente Login:

```
const loginApi = async (user) => {
  try {
    const response = await axios.post(endpoints.login, user);
    localStorage.setItem('usuario', JSON.stringify(response.data));
    navigate('/login/dashboard');
  } catch (error) {
    if (error.response.status === 400 || error.response.status === 404) {
      setShowError( value: true);
      setTimeout( handler: () => {
        setShowError( value: false);
      }, timeout: 2000);
    } else {
      // lidar com outros erros
      console.log(error);
    }
  }
};
```

Figura 14 - Exemplo conexão API

A Figura 14 mostra, como é implementada a comunicação com a API, no qual o pedido é feito através do AXIOS que é uma biblioteca do JavaScript que permite realizar requisições HTTP, ou seja, é possível enviar dados para o servidor e receber dados em formato JSON, XML, HTML, texto ou qualquer outro formato de dados.

A Figura 15 mostra os respetivos endpoints e a API usada durante a fase final do projeto:

```
const API_ROOT = 'https://gowake.dalettech.pt';
export const endpoints = {
  login: `${API_ROOT}/account/login/`,
  createAccount: `${API_ROOT}/account/register/`,
  competitions: `${API_ROOT}/api/competitions/`,
  competitionCreate: `${API_ROOT}/api/competition-create/`,
  competitionsBy: `${API_ROOT}/api/competitions`,
  /* athlete: `${API_ROOT}/api/competition/`, */
  athleteBy: `${API_ROOT}/api/competition`,
  officialBy: `${API_ROOT}/api/competition`,
  eventBy: `${API_ROOT}/api/competition`,
  submitXML: `${API_ROOT}/api/create-all/`,
  heatSystem: `${API_ROOT}/api/generate-heatsystem/`,
  leaderboard: `${API_ROOT}/api/competition`,
};
```

Figura 15 - Endpoints API

Em consequência, caso a requisição seja efetuada com sucesso, ou seja, o status for igual a 200, tem-se a seguinte resposta proveniente da API:

```
{data: {...}, status: 200, statusText: 'OK', headers: AxiosHeaders, config: {...}, ...}
  config: {transitional: {...}, adapter: Array(2), transformRequest: Array(1), transformResponse: Array(1), timeout: 0, ...}
  data:
    email: "bielbene@gmail.com"
    role: "None"
    token: "9422a8a61c49f5387bbf1115a4ccb2208c322e54"
    username: "bielbene"
    [[Prototype]]: Object
  headers: AxiosHeaders {content-length: '117', content-type: 'application/json'}
  request: XMLHttpRequest {onreadystatechange: null, readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequest
    status: 200
    statusText: "OK"
    [[Prototype]]: Object
```

Figura 16 - Response 200

Caso a requisição seja incorreta, status for igual a 400, ter-se-á a seguinte resposta:

```
{data: {...}, status: 400, statusText: 'Bad Request', headers: AxiosHeaders, config: {...}, ...}
  config: {transitional: {...}, adapter: Array(2), transformRequest: Array(1), transformResponse: Array(1), timeout: 0, ...}
  data: {error: 'Wrong Credentials'}
  headers: AxiosHeaders {content-length: '29', content-type: 'application/json'}
  request: XMLHttpRequest {onreadystatechange: null, readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequest
    status: 400
    statusText: "Bad Request"
    [[Prototype]]: Object
```

Figura 17 - Response 400

No entanto, é importante ressaltar que o desempenho e a compatibilidade do React podem variar de acordo com a versão do navegador. Para garantir a compatibilidade com diferentes navegadores e versões, é fundamental estar atualizado e testar a aplicação em diferentes ambientes e, se necessário, utilizar polyfills ou recursos adicionais para garantir a compatibilidade.

A Tabela 3 - Navegadores compatíveis sumariza o que foi dito e mostra os diferentes navegadores assim como as suas compatibilidades relativamente a framework React.

Tabela 3 - Navegadores compatíveis

Navegador	Suporte	Observações
Google Chrome	Suporte completo	O Google Chrome é um dos navegadores mais populares e possui suporte completo para todas as funcionalidades do React.
Microsoft Edge	Suporte completo	O Microsoft Edge é o navegador padrão do Windows 10 e também possui suporte completo para o React.
Mozilla Firefox	Suporte completo	O Mozilla Firefox é outro navegador popular e possui suporte completo para o React.
Opera	Suporte completo	O Opera é outro navegador que tem suporte completo para o React.
Internet Explorer	Suporte limitado	O Internet Explorer tem suporte limitado para o React, principalmente nas versões mais antigas. Algumas funcionalidades do React podem não funcionar corretamente ou precisam de polyfills para funcionar.
Safari	Suporte parcial	O Safari, navegador padrão do sistema operacional macOS, tem suporte parcial para algumas funcionalidades do React. Por exemplo, a função "Array.prototype.includes()" não é suportada nas versões mais antigas do Safari.

5.6 Abrangência

Após a conclusão do projeto, apercebemo-nos do quão intrínseco e interligado são as disciplinas que tivemos durante o percurso académico e o impacto das mesmas foi notório no decorrer deste projeto. As disciplinas abaixo citadas são o reflexo do que achamos ter contribuído e impactado o desenvolvimento do projeto até o seu término:

- Análise e conceção de sistemas – Foi de extrema importância pois facilitou o enquadramento e na identificação do problema assim como o levantamento de requisitos.
- Engenharia de Software – Aprofundou os conhecimentos adquiridos no levantamento de requisitos e a implementação da metodologia Agile contribuiu para a evolução e a colaboração das equipas (Back-End e Front-End). Ou seja, o planeamento adaptativo e o paralelismo contribuíram para a sincronicidade e para a evolução do trabalho entre as equipas.
- Interação Humano Máquina – O conhecimento adquirido por esta unidade curricular ajudou-nos na acessibilidade da aplicação web e na construção do mapa aplicacional.
- Programação Web – É o ponto-chave que inclui o desenvolvimento de interfaces do utilizador, no qual existe a implementação de todas as funcionalidades oriundas da agregação dos conhecimentos adquiridos pelos pontos anteriores.

6 Método e Planejamento

A Figura 18 apresenta o planejamento do projeto desde o seu início até à entrega final. Planejamento este que sofreu várias modificações.

Foi adotada uma metodologia Scrum pois enfatiza a comunicação, colaboração, transparência e a flexibilidade, permitindo que ambas as equipas de desenvolvimento pudessem adaptar-se às mudanças de requisitos e de ambiente, contribuindo assim, para um desenvolvimento de trabalho mais eficaz e eficiente.

Para tal, criou-se cartões, com intuito de simular as tarefas realizadas e a serem implementadas assim como a sua cronologia até a conclusão do respetivo projeto.

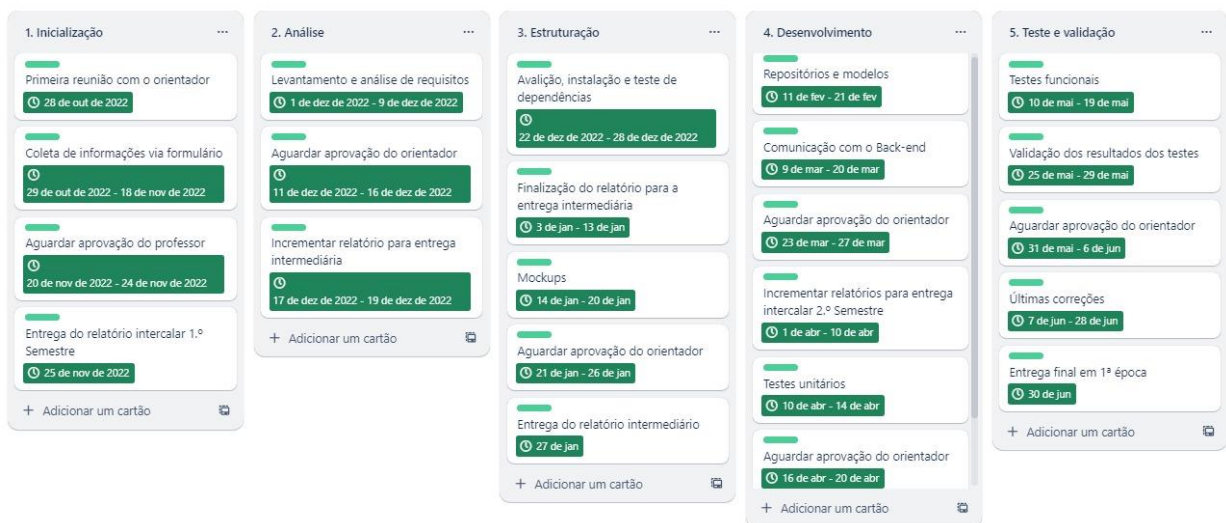


Figura 18 - Planejamento do projeto

A Figura 19, demonstra o cronograma em formato Gantt no qual representa o planejamento do projeto em termos de tempo e atividades. O mesmo é composto por um eixo horizontal que representa o tempo e um eixo vertical que representa as atividades do projeto. Cada atividade é representada por uma barra horizontal, cujo comprimento indica a duração da atividade e a posição indica quando a atividade deve começar e terminar.

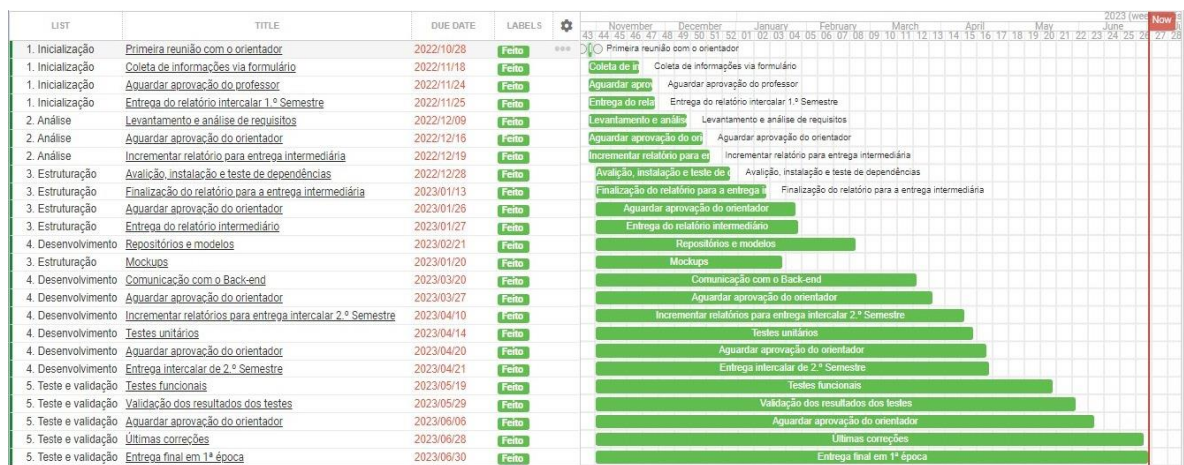


Figura 19 - Cronograma em formato Gantt

7 Resultados

A seguinte Tabela 4 apresenta os casos de testes referentes aos requisitos funcionais mencionados anteriormente.

Tabela 4 - Casos de testes

Caso de Teste	Funcionalidade	Testes	Requisitos	Estado
CT-001	Processo de Login	1,2,3	RF001	Implementado
CT-002	Processo de Registo	4, 5, 6 e 7	RF002	Implementado
CT-003	Recuperar palavra-passe	8 e 9	RF003	Implementado
CT-004	Consultar Regas IWWF Wakeboard	10	RF007	Implementado
CT-005	Consultar Categorias	11	RF007	Não Implementado
CT-006	Leaderboard	12	RF006	Implementado
CT-007	Dashboard	13	N/A	Implementado
CT-008	Upload XML	14,15,16,17, 18	RF004	Implementado
CT-009	Processo para modificar Eventos	19	RF005	Implementado
CT-010	Processo para criar evento	20	RF005	Implementado
CT-011	Interface para atribuir permissões	N/A	RF008	Não Implementado

7.1 – CT-001

- Test Case 1
 - **Given** – Um utilizador abre a aplicação e pretende fazer login, porém não possui registo.
 - **When** – Quando o botão de login é pressionado.
 - **Then** – O sistema deve rejeitar o acesso, pois só mediante uma conta registada e válida que o acesso é permitido.
- Test Case 2
 - **Given** – Um utilizador abre a aplicação e pretende fazer login, porém introduz dados incorretos.
 - **When** – Quando o botão de login é pressionado.
 - **Then** – O sistema deve rejeitar o acesso, pois os dados introduzidos não coincidem com os dados provenientes da API.
- Test Case 3
 - **Given** – Um utilizador abre a aplicação e pretende fazer login com os dados corretos.
 - **When** – Quando o botão de login é pressionado.
 - **Then** – O sistema deve conceder o acesso, disponibilizado as funcionalidades atribuídas ao respetivo utilizador.

7.1 - CT-002

- Test Case 4
 - **Given** – Um utilizador abre a aplicação e pretende efetuar o registo, porém introduz um email inválido.
 - **When** – Quando o botão de registo é pressionado.
 - **Then** – O sistema despoleta uma mensagem de erro, negando o acesso, pois verifica que a integridade do email não é válida.
- Test Case 5
 - **Given** – Um utilizador abre a aplicação e pretende efetuar o registo, porém introduz uma password diferente no campo “Reintroduza a password”.
 - **When** – Quando o botão de registo é pressionado.
 - **Then** – O sistema despoleta uma mensagem de erro, negando o acesso, pois verifica que as passwords não coincidem.
- Test Case 6
 - **Given** – Um utilizador abre a aplicação e pretende efetuar o registo, porém introduz um email já registado na aplicação.
 - **When** – Quando o botão de registo é pressionado.

- **Then** – O sistema despoleta uma mensagem de erro, negando o acesso, pois verifica que o email introduzido já consta na API.
- Test Case 7
 - **Given** – Um utilizador abre a aplicação e pretende efetuar o registo e introduz os dados corretos.
 - **When** – Quando o botão de registo é pressionado.
 - **Then** – O sistema deve conceder o acesso atribuindo, ao respetivo utilizador, uma determinada “role” que ditará as funcionalidades ao que o mesmo terá acesso.

7.1 - CT-003

- Test Case 8
 - **Given** – Um utilizador abre a aplicação e pretende fazer a recuperação da palavra-passe sem estar registado.
 - **When** – Quando a opção de recuperação da palavra-passe é acionada e, posteriormente, o botão de submeter é pressionado.
 - **Then** – O sistema verifica que o email introduzido não se encontra registado e despoleta uma mensagem de erro.
- Test Case 9
 - **Given** – Um utilizador abre a aplicação e pretende fazer a recuperação da palavra-passe.
 - **When** – Quando a opção de recuperação da palavra-passe é acionada e, posteriormente, o botão de submeter é pressionado.
 - **Then** – O sistema verifica que o email introduzido se encontra registado e, envia os dados, para recuperar a palavra-passe, para o respetivo email.

7.1 - CT-004

- Test Case 10
 - **Given** – Um utilizador abre a aplicação e deseja verificar as regras de Wakeboard.
 - **When** – Quando a opção “IWWF Wakeboard Rules” é selecionada pelo utilizador.
 - **Then** – O sistema disponibiliza um sumário das regras mais relevantes e necessárias da modalidade Wakeboard fornecendo a opção de fazer download do documento oficial da IWWF que esmiúça e contem toda a informação.

7.1 - CT-005

- Test Case 11 (**Não implementado**)
 - **Given** – Um utilizador abre a aplicação e deseja verificar Categorias da competição Wakeboard.
 - **When** – Quando a opção “categorias” é selecionada pelo utilizador.
 - **Then** – O sistema disponibiliza uma tabela informativa que exibirá as várias categorias existentes na respetiva modalidade.

7.1 - CT-006

- Test Case 12
 - **Given** – Um utilizador abre a aplicação e deseja consultar as competições
 - **When** – Quando a opção “LeaderBoard” é selecionada pelo utilizador.
 - **Then** – O sistema deve fornecer uma lista de todos os atletas agrupados pelas categorias de um evento específico, apresentando as respetivas notas de cada atleta, bem como permitir a possibilidade de alterar essas notas.

7.1 - CT-007

- Test Case 13
 - **Given** – Um utilizador abre a aplicação e pretende aceder à Dashboard.
 - **When** – Quando o utilizador faz login.
 - **Then** – O sistema disponibiliza uma interface ao utilizador no qual fornece uma lista de todas as competições, possibilitando o mesmo filtrar e pesquisar as competições desejadas.

7.1 - CT-008

- Test Case 14
 - **Given** – Um utilizador abre a aplicação e pretende criar um evento a partir da importação do ficheiro XML válido.
 - **When** – Quando o utilizador faz login e seleciona a opção “Upload XML”.
 - **Then** – O sistema permite ao utilizador criar um evento através da importação de um ficheiro XML, no qual será submetido após a validação das informações pelo Front-End. Após a validação, o evento é enviado para a API para ser novamente validado e, posteriormente, os dados são guardados no Back-End. Em seguida, os dados são chamados e disponibilizados pelo Front-End para serem apresentados ao utilizador.
- Test Case 15
 - **Given** – Um utilizador abre a aplicação e pretende criar um evento a partir da importação do ficheiro XML inválido.
 - **When** – Quando o utilizador faz login e seleciona a opção “Submit”.

- **Then** – O sistema aceita o ficheiro, porém ao tentar submeter ou visualizar as informações do mesmo, é exibido uma mensagem de erro informando que o ficheiro não possui a estrutura correta.
- Test Case 16
 - **Given** – Um utilizador abre a aplicação e pretende criar um evento a partir da importação do ficheiro que não corresponde a um XML.
 - **When** – Quando o utilizador faz login e seleciona a opção “Submit”.
 - **Then** – O sistema não aceita o ficheiro e exibe um alerta informando que o ficheiro selecionado é inválido.
- Test Case 17
 - **Given** – Um utilizador abre a aplicação e pretende criar um evento a partir da importação do ficheiro que já foi submetido corretamente.
 - **When** – Quando o utilizador faz login e seleciona a opção “Submit”.
 - **Then** – O sistema emite um erro proveniente do Back-End a alertar que a respetiva competição / evento já existe.
- Test Case 18
 - **Given** – Um utilizador abre a aplicação e pretende visualizar a informação que existe no ficheiro XML válido.
 - **When** – Quando o utilizador faz login e seleciona a opção “Check”.
 - **Then** – O sistema apresenta uma janela “pop-up” que contém todas as informações contidas no ficheiro.

7.1 - CT-009

- Test Case 19
 - **Given** – Um utilizador abre a aplicação e pretende modificar um evento
 - **When** – Quando o utilizador faz login e seleciona a opção “Edit”.
 - **Then** – O sistema possibilita ao utilizador alocar / editar os oficiais para cada ronda ou heat (bateria) e reatribuir pontuações aos atletas que, outrora, já foram pontuados pelos jurados.

7.1 - CT-010

- Test Case 20
 - **Given** – Um utilizador abre a aplicação e pretende criar um evento de forma manual.
 - **When** – Quando o utilizador faz login e seleciona a opção “Create event”.
 - **Then** – O sistema possibilita ao utilizador criar evento a partir do preenchimento de várias opções disponibilizadas, no qual o mesmo será guardado e validado pelo sistema.

8 Conclusão e trabalhos futuros

8.1 Bibliografia

- [APRT22] Red Hat, <https://www.redhat.com/en/topics/api/what-is-a-rest-api>, acessido em Nov 2022.
- [CICD22] CI/CD, What is CI/CD?, redhat, <https://www.redhat.com/en/topics/devops/what-is-ci-cd>, acessido em Nov 2022.
- [DEISI22] DEISI, Regulamento de Trabalho Final de Curso, Set. 2022.
- [REACT22] REACT, <https://react.dev/learn>, acessido em Nov 2022.
- [ULHT22] Universidade Lusófona, <http://www.ulusofona.pt/>, acessido em Out. 2022.
- [WSFL22] Waterski Scoring Full, <https://www.waterski-softwares.com/download/scoring.html>, acessido em Nov 2022.
- [HTTP22] Hostinger, <https://www.hostinger.com.br/tutoriais/http>, acessido em Nov 2022.

8.2 Anexos

8.2.1 Questionário

Este questionário teve como objetivo avaliar a satisfação na interação com o software Waterski Scoring e identificar os aspetos mais relevantes a serem considerados em termos de desempenho, processos e design de interação, com o objetivo de realizar melhorias.

Tabela 5 - Utilizadores

	Male	Female
Your gender *	①	①

Tabela 6 - Grau de usabilidade

	Strongly disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
Overall, I am satisfied with the system	①	①	①	①	①
It was simple to use this system	①	①	①	①	①

The information (such as online help, on-screen messages) provided with this system is clear	①	①	①	①	①
This interface is slow and takes too long to retrieve the info I want	①	①	①	①	①
The system provides easy readability and good contrast	①	①	①	①	①
I get lost while using the application and it's difficult to find things	①	①	①	①	①
The interface of this system is pleasant	①	①	①	①	①
This system has all the functions and capabilities I expect it to have	①	①	①	①	①

8.2.2 Progresso do trabalho

Nesta entrega final, houve as seguintes modificações e melhorias:

- Correção de legendas das figuras.
- Uma melhor calendarização e cronograma.
- Efetuada a comunicação do Back-End com o Front-End.
- Correção de erros gramaticais e incoerências de algumas palavras.
- Uma arquitetura e ferramentas utilizadas mais bem estruturadas e concisas.
- Implementação das páginas que, outrora, não haviam sido implementadas na entrega anterior.

8.3 Glossário

API	Application programming Interface
CSS	Cascading Style Sheets (Folha de Estilo em Cascatas)
DEISI	Departamento de Engenharia Informática e Sistemas de Informação
HTML	Linguagem de Marcação de HiperTexto
IWWF	International Waterski & Wakeboard Federation
JS	JavaScript
LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
PDF	Portable Document Format (Formato de Documento Portátil)
TFC	Trabalho Final de Curso
UI	User Interface (Interface do Utilizador)
UX	User Experience (Experiência do Utilizador)
XML	Extensible Markup Language