



UNIVERSIDADE  
LUSÓFONA

# Plataforma de gestão de eventos – mobile app

## Trabalho Final de curso

Relatório Intercalar 2º Semestre

Aluna: Catarina Moita

Orientador: Rodrigo Correia

Coorientador: Bruno Cipriano

Trabalho Final de Curso | LEI | 23/04/2021

[www.ulusofona.pt](http://www.ulusofona.pt)

## **Direitos de cópia**

*Plataforma de gestão de eventos – mobile app*, Copyright de *Catarina Moita, Rodrigo Correia & Bruno Cipriano*, ULHT.

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

---

## Resumo

No presente Trabalho Final de Curso (TFC) pretende-se desenvolver uma aplicação móvel que integre um sistema capaz de suportar um clube de corrida de uma empresa.

Para a concretização deste trabalho é fundamental contextualizar o tema e identificar o problema concreto em estudo. Assim, são apresentadas as circunstâncias a que levaram a esta proposta a ser realizada, como este projeto parte de um problema real de uma organização e quais as soluções apresentadas para o resolver. Partindo destes conhecimentos, ao longo do relatório, serão produzidas conclusões, na medida de validar o produto final obtido, face ao proposto na primeira entrega ao longo de todo o trabalho.

Posteriormente, parte-se para a análise da viabilidade e pertinência. É escrutinada a capacidade do trabalho poder ser continuado e implementado no dia a dia, mesmo após a conclusão da avaliação para fins académicos e as vantagens que traria o seu desenvolvimento.

A fim de garantir o sucesso do trabalho é imprescindível a identificação e análise de requisitos validados pelas partes interessadas e evidenciar o contributo que têm na fase da descrição detalhada da solução proposta. Será ainda dada especial atenção às tecnologias que se vão utilizar durante a implementação, o porquê destas serem as opções mais viáveis e os fatores técnicos que contribuíram para a construção da solução funcional.

De modo a enquadrar a solução proposta face a alternativas e potenciais concorrentes do mercado é feito o *benchmarking*, comparando diretamente a solução idealizada com outros produtos já existentes de forma a que esta apresente diferenciais inovadores.

Será ainda dedicado um capítulo ao impacto da interdependência de todos estes pontos com a realização de um plano de testes e como estes asseguram o sucesso da aplicação móvel no meio empresarial a que se destina.

Por fim, é apresentado um plano e cronograma do trabalho com todos os objetivos a ser cumpridos até à entrega final do TFC.

**Palavras-chave:** aplicação móvel, atividades desportivas, gestão de eventos, flutter

## **Abstract**

The purpose of this Final Year Project (TFC) is to develop a mobile application that integrates a system capable of supporting a company's running club.

In order to carry out this work, it is essential to contextualize the theme and identify the concrete problem under study. Thus, the circumstances that led to this proposal to be carried out are presented, how this project starts from a real problem of an organization and what solutions are presented to solve it. Based on this knowledge, throughout the report, conclusions will be produced, to the extent of validating the final product obtained, compared to that proposed in the first delivery throughout the entire work.

Subsequently, we proceed to the analysis of feasibility and pertinence. The ability of the work to be continued and implemented on a daily basis is scrutinized, even after the conclusion of the evaluation for academic purposes and the advantages that its development would bring.

In order to guarantee the success of the work, it is essential to identify and analyze requirements validated by the interested parties and to highlight the contribution they have in the detailed description of the proposed solution. Special attention will also be given to the technologies that will be used during the implementation, why these are the most viable options and the technical factors that contributed to the construction of the functional solution.

To fit the proposed solution in the face of alternatives and potential competitors in the market, benchmarking is done, directly comparing the idealized solution with other existing products so that it presents innovative differentials.

A chapter will also be dedicated to the impact of the interdependence of all these points with the realization of a test plan and how they ensure the success of the mobile application in the intended business environment.

Finally, a work plan and schedule is presented with all the objectives to be met until the final delivery of the TFC.

**Keywords:** mobile application, sports events, event management, flutter

---

# Índice

Resumo.....	iii
Abstract .....	iv
Índice.....	v
Lista de Figuras .....	vii
Lista de Tabelas .....	viii
1 Identificação do Problema .....	1
2 Viabilidade e Pertinência.....	3
3 Levantamento e Análise de Requisitos .....	5
3.1 Requisitos Funcionais .....	5
3.1.1 Estado de implementação.....	10
3.2 Requisitos Não Funcionais .....	10
4 Solução Proposta.....	11
4.1 Funcionalidades.....	11
4.2 Tecnologias.....	13
4.3 Arquitetura.....	15
4.4 Protótipo Axure .....	16
4.5 Implementação .....	17
4.5.1 Responsive.....	17
4.5.2 Navegação .....	17
4.6 Protótipo Flutter.....	19
4.7 Código Fonte .....	27
4.8 Vídeo .....	27
5 Benchmarking.....	28
6 Plano de testes e validação .....	32
7 Calendário .....	33
Bibliografia .....	35
Anexo 1 - Implementação do LayoutBuilder() e MediaQuery.of() .....	38
Anexo 2 - Implementação do Provider .....	39
Glossário.....	40



---

# Lista de Figuras

Figura 1 - Gráfico das inscrições e provas ao longo dos anos .....	3
Figura 2 - Mapa Aplicacional .....	12
Figura 3 - Quota de mercado mundial dos sistemas operativos móveis em setembro de 2020 .....	14
Figura 4 - Arquitetura do sistema geral.....	15
Figura 5 - Ecrã login Axure .....	16
Figura 6 - Ecrã principal Axure.....	16
Figura 7 – Menu administrador Axure.....	16
Figura 8 – Ecrã estatísticas Axure .....	16
Figura 9 - Arquitetura MVVM .....	18
Figura 10 - Splash screen .....	19
Figura 11 - Ecrã login .....	20
Figura 12 - Ecrã principal .....	20
Figura 13 - Menu administrador.....	20
Figura 14 - Menu membro.....	20
Figura 15 - Campo de pesquisa .....	21
Figura 16 - Ecrã detalhe de evento.....	21
Figura 17 - Inscrição e cancelamento de evento .....	22
Figura 18 - Mensagem de erro prazo .....	22
Figura 19 - Mensagem de erro plafond.....	23
Figura 20 - Ecrã aprovação de eventos.....	23
Figura 21 - Ecrã aprovação de membros.....	23
Figura 22 - Rejeitar evento .....	24
Figura 23 - Ecrã estatísticas .....	24
Figura 24 - Ecrã filtros.....	25
Figura 25 - Ecrã estatísticas após aplicação de filtros .....	26
Figura 26 - Registo de tempos .....	26
Figura 27 - Aplicação Eventsport .....	28
Figura 28- Aplicação Running Lisboa .....	29
Figura 29 - Aplicação Pronto a Correr .....	29
Figura 30 - Aplicação Run Races .....	30
Figura 31 - Aplicação RamRun .....	30
Figura 32 - Calendário de testes .....	32
Figura 33 - Gráfico de Gant do plano de trabalho proposto.....	33

## **Lista de Tabelas**

Tabela 1 - User Story 1	5
Tabela 2 - User Story 2	6
Tabela 3 - User Story 3	6
Tabela 4 - User Story 4	6
Tabela 5 - User Story 5	7
Tabela 6 - User Story 6	7
Tabela 7 - User Story 7	7
Tabela 8 - User Story 8	8
Tabela 9 - User Story 9	8
Tabela 10 - User Story 10	8
Tabela 11 - User Story 11	9
Tabela 12 - User Story 12	9
Tabela 13 - User Story 13	9
Tabela 14 - User Story 14	10
Tabela 15 - Funcionalidades da aplicação móvel de acordo com o tipo de utilizador final	11
Tabela 16 - Comparação da aplicação a ser desenvolvida com aplicações concorrentes	31



# 1 Identificação do Problema

No ano letivo 2019/2020 foi feita uma proposta no âmbito da unidade curricular Trabalho Final de Curso cujo objetivo era desenvolver um sistema de gestão de eventos, baseado em tecnologia web, para ser aplicado numa Pequena e Média Empresa (PME) portuguesa. Este sistema, realizado pelos alunos a que o tema foi atribuído, consiste numa plataforma para uso interno por parte da organização, onde os membros (funcionários da empresa associados a um *plafond* predefinido) têm a possibilidade de se inscreverem em eventos desportivos, tais como, corridas e caminhadas, que sejam do seu interesse.

A sua implementação veio contribuir para uma maior centralização dos dados e facilitar o esforço de gestão do clube que, outrora dependia de inúmeros ficheiros Excel para guardar todas as informações dos funcionários registados como membros e da plataforma externa Doodle [1] como meio para efetuar as inscrições em cada evento.

Contudo carecia de uma aplicação móvel que apoiasse esta componente web. Trata-se, portanto, de uma necessidade real de uma empresa.

Face ao problema exposto, o produto final deste trabalho é uma aplicação móvel complementar ao sistema web já existente, para uso interno de uma *software house* nacional, que permite a inscrição dos seus funcionários em atividades desportivas, fazer a gestão de membros e eventos e oferecer um serviço mais personalizado com o registo dos tempos pessoais de conclusão de provas e a opção de acompanhar a sua evolução estatística.

O sistema desenvolvido teve por base a proposta que serviu de prelúdio ao primeiro momento de avaliação mas, como em qualquer processo natural de desenvolvimento de projetos, sofreu algumas alterações resultantes de fatores como a evolução de conhecimentos da temática, alteração de prioridades dos intervenientes e algumas inconsistências encontradas.

Comparativamente ao idealizado inicialmente, não foi adotada a funcionalidade de registo do utilizador. Por outro lado, acrescentou-se a opção de filtrar os tempos de conclusão de prova por tipo de atividade, distância percorrida e/ou intervalo de datas, o indicador dos totais de tempo e distância de provas e a opção de um administrador poder utilizar a barra de pesquisa para encontrar, pelo nome, membros e eventos pendentes de aprovação.



## 2 Viabilidade e Pertinência

Mas porquê uma aplicação móvel?

Como se pode observar na Figura 1, verifica-se que durante o período dos anos 2017 a 2019<sup>1</sup>, apesar do número de provas disponíveis ter aumentado aproximadamente 47,8% em relação ao primeiro ano desta iniciativa, assistiu-se a uma queda de cerca 60% do número de inscrições.

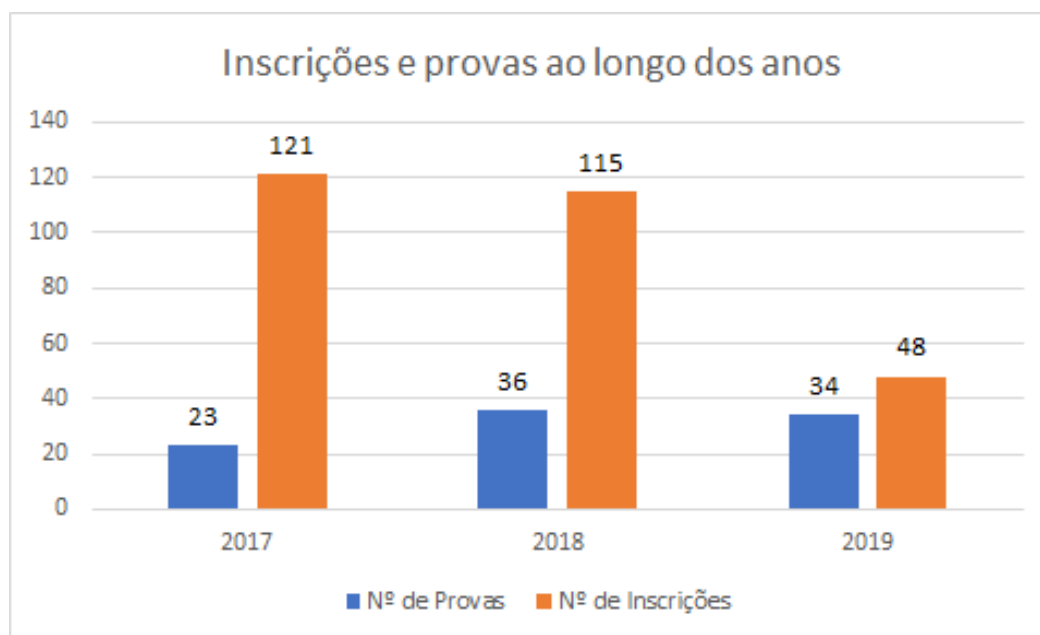


Figura 1 - Gráfico das inscrições e provas ao longo dos anos

A capacidade de facultar uma nova experiência aos utilizadores, através deste dispositivo, em qualquer local a qualquer hora é o cerne das aplicações móveis. A ideia de ter a plataforma de gestão de eventos em formato móvel resulta numa maior conveniência tanto para o administrador que pode gerir todos os eventos sempre que o entender, como para o utilizador na inscrição de provas, por exemplo, sem ter de andar constantemente com um computador atrás sempre que se quer realizar alguma ação.

Inclusive, o utilizador tem a possibilidade de receber notificações não intrusivas de novos eventos disponíveis sem ter de aceder ao email da empresa, de registar o tempo em que concluiu a prova e, com isto, consultar estatísticas personalizadas (prova mais rápida, mais lenta, entre outras) diretamente no telemóvel.

De destacar que a aplicação móvel oferece a opção de fazer a integração com as redes sociais (Twitter, Instagram, Facebook). Significa isto que, os funcionários da empresa, podem partilhar nas diversas redes sociais em que provas vão participar e associar o nome da empresa às publicações. Neste sentido, qualquer pessoa que se cruza-se com estas publicações poderia ter interesse em saber mais sobre a organização, contribuindo para a difusão da mesma.

Uma vez definida a pertinência do projeto deve-se agora estudar a sua viabilidade. Durante o desenvolvimento de qualquer aplicação depara-se com várias questões mas algo tem de ser tido sempre em conta: o produto final tem de funcionar no maior número de dispositivos

---

<sup>1</sup> Derivado da pandemia não tem havido participação em provas, logo não existe registo de dados mais recentes

possíveis, tornando-o assim mais abrangente. A decisão de implementar a aplicação com suporte multiplataforma foi crucial neste aspeto (no capítulo correspondente à Solução Proposta será dado a conhecer como é que este conceito se materializou) .

O facto deste projeto não só incluir muitos dos conteúdos programáticos abordados ao longo do curso de Engenharia Informática e relevantes para a conclusão do percurso académico, mas também de reunir todas as condições para o seu resultado final ser usado num contexto real e até continuado no futuro (acrescentar ou mudar funcionalidades consoante as necessidades da empresa) mostra a sua aplicabilidade e viabilidade.

Estes fatores serão fortemente postos à prova na fase de testagem. Colocar o *software* num ambiente real a ser testado pelas partes interessadas será imprescindível na confirmação de que a aplicação móvel desenvolvida representa um passo para a solução do problema proposto no presente TFC, contornando a tendência que se tem observado até agora.

### 3 Levantamento e Análise de Requisitos

O processo de identificação detalhado de requisitos e, posteriormente, a sua análise, no desenvolvimento do sistema de modo a responder aos problemas e expectativas da organização a que se destina é de extrema importância para o sucesso do projeto. Pretende-se desta forma, evitar futuros erros que vão desde atrasos na conceção da aplicação, consequente da elaboração de requisitos pouco realistas dado ao tempo disponível e objetivos calendarizados, até a casos mais graves como o produto final não corresponder ao idealizado pelo cliente.

Estes requisitos terão, portanto, de ser compatíveis mediante as necessidades e critérios comunicados durante a fase de elicitação, ter em conta os conflitos e dependências que possam existir entre eles em prol da conservação da integridade de cada um e quais os seus níveis de prioridade. Dada a sua finalização e aprovação, este documento servirá como base para a implementação da aplicação móvel.

Foram identificados dois tipos de requisitos: funcionais e não funcionais.

Optou-se por realizar os requisitos funcionais na forma de *user stories*, aplicados no desenvolvimento Agile [2]. “No desenvolvimento de software e gestão de produto, uma *user story* é uma descrição informal em linguagem natural de uma ou mais características do sistema de software” [3]. Articulam na perspetiva do utilizador final o “para quem”, “o quê” e o “porque” de forma simples e eficaz, poupando tempo na escrita exaustiva de requisitos e dando ênfase ao que o cliente realmente quer. Este método vai traduzir-se num benefício a longo prazo perante um cenário de continuidade do projeto além do TFC, uma vez que, a sua versatilidade e a preocupação que existe em evitar entrar em muitos detalhes prematuramente vão permitir a exploração de novas funcionalidades e evitar a restrição de possíveis soluções no momento de implementação de código.

No decorrer deste trabalho surgiram assim quatorze *user stories*, onde dez são do ponto de vista dos dois tipos de utilizadores e os restantes quatro restritos ao utilizador do tipo administrador.

Todos os requisitos terão uma indicação sobre o seu estado atual (cumprimento parcial ou integral) e, se aplicável, se o requisito foi modificado, retirado ou acrescentado, comparativamente ao inicialmente proposto na avaliação anterior.

#### 3.1 Requisitos Funcionais

Requisitos que descrevem o comportamento que a aplicação deve ter, quais as suas características e funcionalidades.

Tabela 1 - User Story 1

User Story 1
<p><b>Como</b> membro e administrador</p> <p><b>Consigo</b> fazer <i>login</i></p> <p><b>De forma</b> a poder usar as funcionalidades da aplicação</p>
<p><b>Critérios de Aceitação</b></p> <ol style="list-style-type: none"> <li>1. Consigo ver um formulário com os campos para introdução do nome de utilizador e palavra-passe</li> <li>2. A palavra-passe nunca será visível</li> <li>3. Ao introduzir dados válidos correspondentes a um utilizador preexistente na Base de Dados, sou redirecionado para o ecrã principal da aplicação</li> </ol>

Tabela 2 - User Story 2

User Story 2
<b>Como</b> membro e administrador <b>Consigo</b> fazer <i>logout</i> <b>De forma a</b> poder terminar a minha sessão na aplicação
<b>Critérios de Aceitação</b> 1. Saio da minha sessão e sou redirecionado para a o ecrã de <i>login</i>

Tabela 3 - User Story 3

User Story 3
<b>Como</b> membro e administrador <b>Consigo</b> ter acesso à lista de provas disponíveis <b>De forma a</b> poder inscrever-me nas que me interessam
<b>Critérios de Aceitação</b> 1. Consigo ver uma lista com as provas a se realizar em data futura 2. Ao clicar numa prova, posso aceder aos detalhes completos da mesma

Tabela 4 - User Story 4

User Story 4
<b>Como</b> membro e administrador <b>Consigo</b> inscrever-me numa prova <b>De forma a</b> poder participar no evento
<b>Critérios de Aceitação</b> 1. Consigo ver todos os detalhes respeitantes à prova (nome, tipo de atividade, data a realizar-se, data limite de inscrição, localização e custo) 2. Para que a inscrição seja aceite, o custo do evento tem de ser inferior ou igual ao valor disponível no meu <i>plafond</i> 3. Se o prazo de inscrição já tiver sido ultrapassado, o registo no evento não é válido
<b>Notas:</b> Requisito modificado - o ponto 3 foi acrescentado

Tabela 5 - User Story 5

User Story 5
<b>Como</b> administrador <b>Consigo</b> aceder à lista de membros <b>De forma a</b> poder aprovar ou rejeitar novos utilizadores que se tenham registado no sistema
<b>Critérios de Aceitação</b> <ol style="list-style-type: none"><li>1. Consigo ver uma lista de novos utilizadores que ainda não foram aprovados</li><li>2. Tenho a capacidade de aprovar ou rejeitar o membro através dos respetivos botões</li></ol>

Tabela 6 - User Story 6

User Story 6
<b>Como</b> administrador <b>Consigo</b> aceder à lista de eventos <b>De forma a</b> poder aprovar ou rejeitar novos eventos que tenham sido criados no sistema
<b>Critérios de Aceitação</b> <ol style="list-style-type: none"><li>1. Consigo ver uma lista de novos eventos que ainda não foram aprovados</li><li>2. Tenho a capacidade de aprovar ou rejeitar o evento através dos respetivos botões</li></ol>

Tabela 7 - User Story 7

User Story 7
<b>Como</b> membro e administrador <b>Consigo</b> pesquisar uma prova <b>De forma a</b> poder encontrá-la mais facilmente
<b>Critérios de Aceitação</b> <ol style="list-style-type: none"><li>1. No campo da pesquisa da lista de eventos, ao introduzir o nome da prova, obtenho o evento correspondente</li></ol>

Tabela 8 - User Story 8

User Story 8
<b>Como</b> administrador <b>Consigo</b> pesquisar um membro pendente de aprovação <b>De forma a</b> poder encontrá-lo mais facilmente
<b>Critérios de Aceitação</b> <ol style="list-style-type: none"><li>1. No campo da pesquisa da lista de membros pendentes de aprovação, ao introduzir o nome do utilizador, obtenho o membro correspondente</li></ol>

Tabela 9 - User Story 9

User Story 9
<b>Como</b> administrador <b>Consigo</b> pesquisar uma prova pendente de aprovação <b>De forma a</b> poder encontrá-la mais facilmente
<b>Critérios de Aceitação</b> <ol style="list-style-type: none"><li>1. No campo da pesquisa da lista de eventos pendentes de aprovação, ao introduzir o nome da prova, obtenho o evento correspondente</li></ol>

Tabela 10 - User Story 10

User Story 10
<b>Como</b> membro e administrador <b>Consigo</b> ver o meu <i>plafond</i> <b>De forma a</b> ter conhecimento do valor que tenho disponível para gastar
<b>Critérios de Aceitação</b> <ol style="list-style-type: none"><li>1. O <i>plafond</i> é visível ao longo da aplicação</li><li>2. Sempre que me inscrevo numa prova, o custo desta é debitado no <i>plafond</i> e o valor é atualizado</li></ol>



Tabela 11 - User Story 11

User Story 11
<p><b>Como</b> membro e administrador</p> <p><b>Consigo</b> partilhar nas minhas redes sociais os eventos que irei atender</p> <p><b>De forma a</b> poder efetuar publicações para os meus seguidores</p>
<p><b>Critérios de Aceitação</b></p> <ol style="list-style-type: none"> <li>1. Consigo ver um ícone que, ao clicar, irá me facultar uma caixa de texto <i>default</i></li> <li>2. Submeto a publicação nas redes sociais através de um botão</li> </ol>

Tabela 12 - User Story 12

User Story 12
<p><b>Como</b> membro e administrador</p> <p><b>Consigo</b> consultar as minhas estatísticas</p> <p><b>De forma a</b> ser capaz de observar a minha evolução em todas as provas que participei</p>
<p><b>Critérios de Aceitação</b></p> <ol style="list-style-type: none"> <li>1. Consigo ver a prova que concluí mais rapidamente</li> <li>2. Ver a prova que concluí mais lentamente</li> <li>3. Ver o total de distância percorrida</li> <li>4. Ver o total de tempos</li> <li>5. Ter acesso ao gráfico linear da variável tempo, em horas, e a data de conclusão da prova</li> </ol>
<p><b>Notas:</b> Requisito modificado – ponto 3 e 4 acrescentados</p>

Tabela 13 - User Story 13

User Story 13
<p><b>Como</b> membro e administrador</p> <p><b>Consigo</b> adicionar o tempo em que terminei uma prova</p> <p><b>De forma a</b> poder observar a evolução estatística de todas as provas</p>
<p><b>Critérios de Aceitação</b></p> <ol style="list-style-type: none"> <li>1. O formulário deve validar que apenas tempos positivos e não nulos sejam permitidos</li> </ol>

Tabela 14 - User Story 14

User Story 14
<b>Como</b> membro e administrador <b>Consigo</b> filtrar os tempos em que terminei uma prova <b>De forma a</b> poder observar a evolução estatística de todas as provas segundo um dado parâmetro
<b>Critérios de Aceitação</b> 1. Consigo filtrar os resultados por distância percorrida, pelo tipo de prova (corrida ou caminhada) e/ou intervalo de datas
<b>Notas:</b> Requisito acrescentado

### 3.1.1 Estado de implementação

- Na componente móvel está tudo implementado em termos de UI e os requisitos foram cumpridos na íntegra;
- Falta fazer as ligações ao servidor/*web services*. De momento são apresentados ao utilizador dados *hardcoded* em JSON;

## 3.2 Requisitos Não Funcionais

Requisitos que especificam os atributos de qualidade do sistema. Têm um papel tão crítico no sucesso da aplicação como os requisitos funcionais, visto que afetam diretamente a experiência do utilizador.

- a) A aplicação irá assegurar que apenas os utilizadores devidamente registados na Base de Dados do sistema e aprovados por um administrador possam efetuar o *login* com sucesso e ter acesso às funcionalidades da plataforma;
- b) A aplicação deverá ser compatível com todas as versões suportadas [4] dos dispositivos Android com SO Android Jelly Bean, v16, 4.1.x ou mais recente e *hardware* com processador ARM e dispositivos iOS com SO iOS 8 ou mais recente e *hardware* iPhone 4S ou mais recente;
- c) A Experiência de Utilizador (UX) e Interface de Utilizador (UI) do sistema deverá estar em concordância com as práticas de usabilidade das aplicações móveis [5] ;

## 4 Solução Proposta

### 4.1 Funcionalidades

Ao fim de desenvolver a solução mais adequada para o problema em estudo e coerente com os requisitos expostos no capítulo anterior, foram implementadas as funcionalidades consoante a Tabela 15.

**Tabela 15 - Funcionalidades da aplicação móvel de acordo com o tipo de utilizador final**

Funcionalidades	Membro	Administrador
Login	✓	✓
Aprovar membro		✓
Aprovar evento		✓
Procurar membro pendente de aprovação		✓
Procurar evento pendente de aprovação		✓
Procurar evento	✓	✓
Inscrever em evento	✓	✓
Registar o tempo de conclusão de prova*	✓	✓
Consultar estatísticas sobre tempos de prova: melhor tempo e pior tempo*	✓	✓
Consultar estatísticas sobre totais de provas: total de tempos e de distância*	✓	✓
Filtrar estatísticas: distância*	✓	✓
Filtrar estatísticas: tipo de atividade*	✓	✓
Filtrar estatísticas: intervalo de datas*	✓	✓
Partilhar nas redes sociais* <sup>2</sup>	✓	✓

Considerando a Tabela 15, foi proposto converter para a vertente móvel a generalidade das funcionalidades de utilizador e algumas de administrador já definidas na componente web preexistente e outras opções extras anteriormente visadas na secção Viabilidade e Pertinência.

Esta segmentação de funcionalidades para cada tipo de utilizador confere à aplicação a capacidade de se adaptar a cada contexto e situação. Tanto o membro como o administrador podem realizar o *login* através das credenciais que introduziram durante o registo efetuado na

<sup>2</sup> Funcionalidades exclusivas do sistemas *mobile*

plataforma web e, consoante estas mesmas credenciais, será apresentado um conjunto de funcionalidades específicas a cada utilizador.

Enquanto membro, podemos aceder à lista completa de eventos, inscrever-nos tendo em conta o saldo do *plafond* disponível e procurá-los por características, como o nome, para facilitar a obtenção imediata dos resultados desejados, reduzindo assim a complexidade de navegação e poupando tempo ao utilizador. O membro pode ainda registar os tempos em que concluiu uma determinada prova, consultar as estatísticas resultantes da incorporação destes dados, aplicar filtros, e partilhar os eventos em que vai participar nas suas redes sociais. No registo de tempos de conclusão de provas optou-se por não proceder a nenhuma verificação da veracidade do tempo introduzido tendo em conta fatores como a distância da prova e o mínimo humanamente atingível, pois tal poderia ser perceptível pelo utilizador como algo hostil. Atendendo que os participantes são funcionários da mesma empresa e muitos vão aos eventos pelo convívio, será deixado ao critério de cada um a verificação dos tempos que introduzem, sendo apenas validado se os valores são positivos e não nulos.

Já enquanto administrador, temos acesso às mesmas funcionalidades que o membro e, adicionalmente, assumir o papel de aprovar ou rejeitar membros e/ou eventos. O processo de pesquisa por nome (quer de membros como de eventos) foi igualmente implementado, mantendo o nível de sofisticação por toda a aplicação. De momento, tanto o registo de novos utilizadores como a criação de eventos caberá exclusivamente à plataforma web.

Esquemáticamente expõe-se na Figura 2 a estrutura da solução apresentada na forma de mapa aplicacional para o membro (a laranja) e para o administrador (a verde).

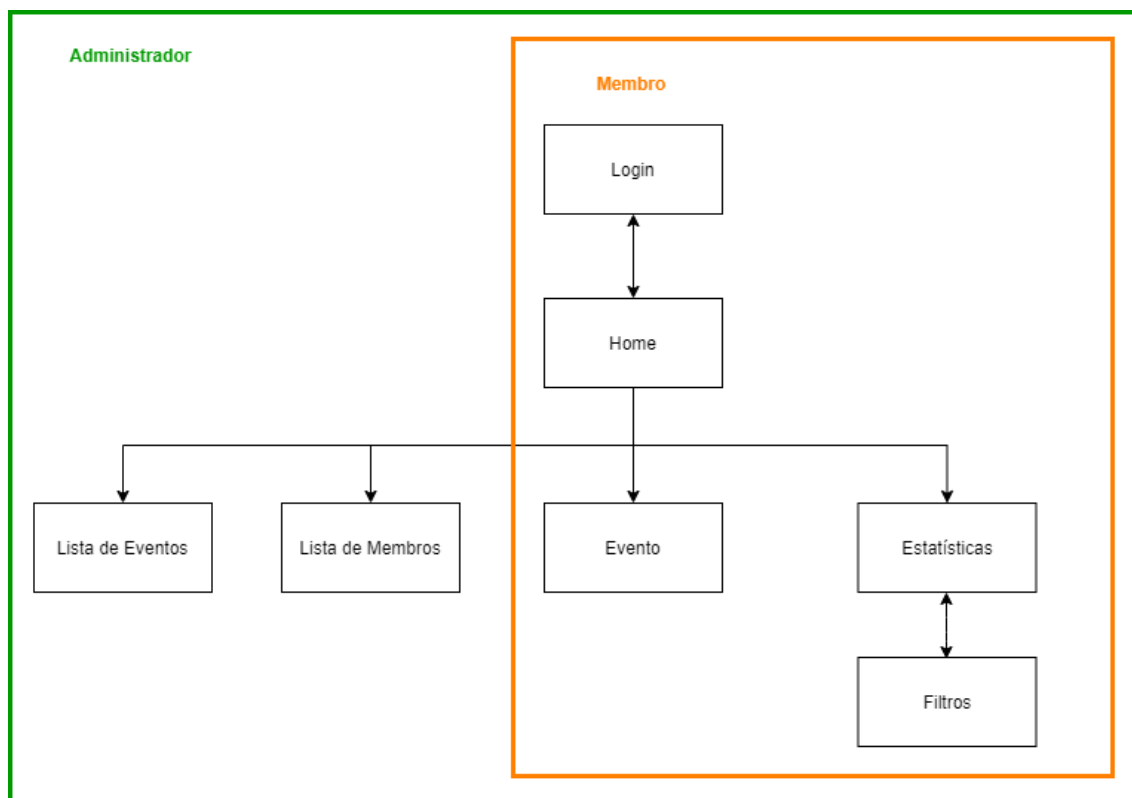


Figura 2 - Mapa Aplicacional

De forma a abarcar todas as ideias fundamentais, a aplicação tem cinco ecrãs para o membro e sete ecrãs para o administrador. É apresentado um *splash screen* inicial em ambos os casos que, de seguida, exibirá o ecrã de *login*. Feito o *login*, surge o ecrã principal com a lista de todos os eventos disponíveis. Ao interagir com um dos eventos, os utilizadores são direcionados para o ecrã do evento em questão que contém todos os detalhes associados (nome, tipo de

atividade, localização, custo, data da prova e data limite de inscrição) e no qual podem inscrever-se e usufruir da opção de partilha nas redes sociais.

Ainda alusivo ao conjunto de ecrãs comuns aos dois tipos de utilizadores, tem-se o ecrã das estatísticas. Este ecrã será o foco do registo de tempos de término de provas e, face aos valores introduzidos, gera estatísticas sob forma de gráfico. Todos estes resultados podem ser filtrados por distância (5km, 10km, 21km e 42km), por tipo de prova (corrida ou caminhada) e por intervalo de datas.

Os ecrãs com a lista de membros e eventos (específicos do administrador) são dedicados à aprovação ou rejeição de membros e eventos, respetivamente, e fará uso da ferramenta de pesquisa abordada em pontos anteriores.

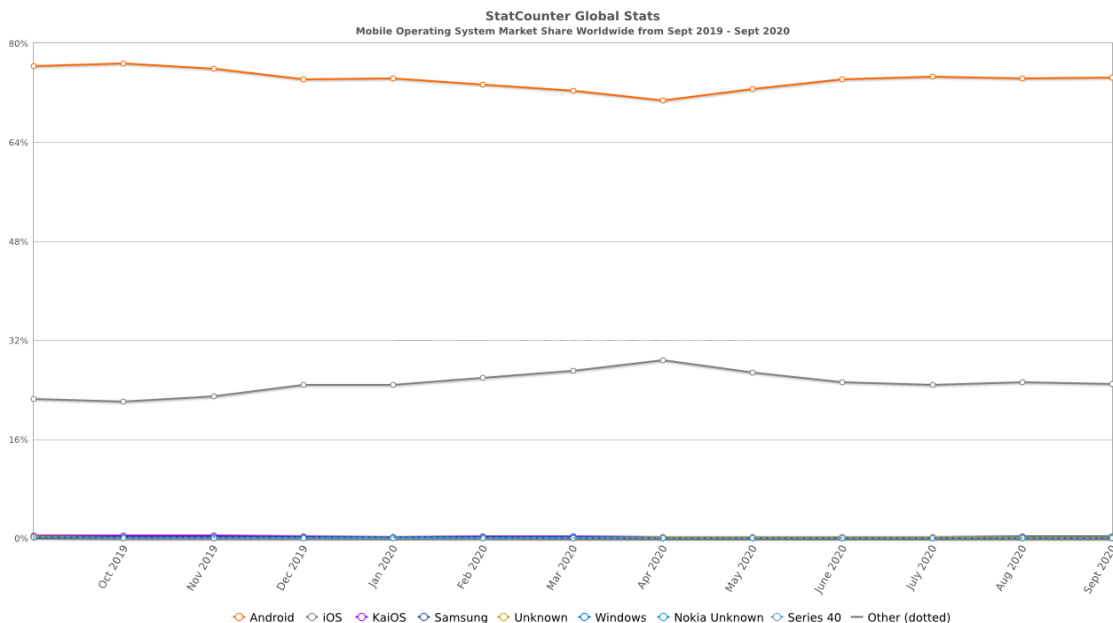
## 4.2 Tecnologias

Em vista da solução proposta, no que toca às tecnologias a utilizar para a sua concretização tem-se:

- Flutter
- JSON

Através do mesmo código base o Flutter [6], uma *framework open source* criada pela Google, consegue construir aplicações quer seja para ambientes móveis (Android, iOS) como para web e desktop (Windows, macOS, Linux). O facto do Flutter ter esta flexibilidade e poupar o programador de reescrever o código para que a aplicação se possa adaptar a cada particularidade das diferentes plataformas traz vastos benefícios na gestão de tempo. A título de exemplo, se a empresa quisesse desenvolver a aplicação móvel tanto para Android como para iOS, não teria a necessidade de contratar duas equipas diferentes para cada sistema, contribuindo assim para a diminuição dos custos de produção ou ter uma equipa responsável pelo desenvolvimento para ambos os sistemas operativos (SO), aumentando a quantidade de tempo gasto na migração de plataformas que, como alternativa, pode ser investido na otimização da aplicação. É de extrema importância ter sempre o futuro em conta e pensar na estrutura de um programa de modo a que seja de fácil manutenção e mudança.

No caso deste projeto, o foco do Flutter será na construção de uma aplicação móvel que execute em diferentes SO. De acordo com o Statcounter [7], em setembro de 2020 o Android tinha uma taxa de ocupação no mercado de 74,44% e o iOS de 24,98% (Figura 3). Com base nestes dados pode-se assumir que público alvo da aplicação em questão está concentrado nestes SO, evitando o cenário à partida ideal de construir uma aplicação para todas as plataformas referidas na Figura 3 mas que se tornaria contraproducente.



**Figura 3 - Quota de mercado mundial dos sistemas operativos móveis em setembro de 2020**

Para além da sua versatilidade, o Flutter distingue-se por ter uma performance equivalente a de uma aplicação nativa, já que não depende de intermediários para o código ser interpretado, por ser muito intuitivo, o que promove o rápido desenvolvimento, de usar a sua própria renderização de *widgets* dedicadas para cada SO e por oferecer a possibilidade de personalizar a UI de forma atrativa, respeitando as boas práticas de design, graças à linguagem de programação Dart [8] e ao mecanismo de renderização de alta performance Skia [9] .

Tudo isto mostra o Flutter como sendo uma mais valia e a escolha acertada para o desenvolvimento deste trabalho.

Antes de explorar o conceito de JSON [10] , deve-se definir a noção de *web services*. Os *web services* funcionam como um agregador de serviços, que comunicam com um servidor remoto através de tecnologias e protocolos Web da forma mais eficiente, eficaz e segura (não há acesso direto à Base de Dados) que os dispositivos móveis exigem. Oferecem uma enorme flexibilidade e dinamismo, atuando nos mais diversos sistemas operativos e *hardware*, uma vez que cada aplicação pode ter a sua própria linguagem, que é mais tarde traduzida num formato intermediário. Neste caso será utilizado o formato JSON, uma sintaxe aplicada à troca e armazenamento de dados, usado em *web services* REST, que tem a particularidade de ser facilmente escrito e entendido pelo ser humano.

Mais importante ainda, o Flutter tem bibliotecas próprias para o desenvolvimento de *web services* REST baseados em JSON, coadunando estas duas tecnologias.

Caso seja identificada a necessidade de ser implementado um novo *web service* para dar resposta a requisitos específicos da aplicação móvel pode ser utilizado a linguagem de programação multiparadigma Common-Lisp [11] adotada pelos alunos do ano passado durante a construção do lado do servidor, uma vez que os programadores da empresa alvo trabalham quase exclusivamente com esta linguagem.

### 4.3 Arquitetura

Em termos esquemáticos, a Figura 4 mostra a arquitetura do sistema em geral e como a componente móvel implementada (a azul) se liga ao sistema já existente (preto e branco):

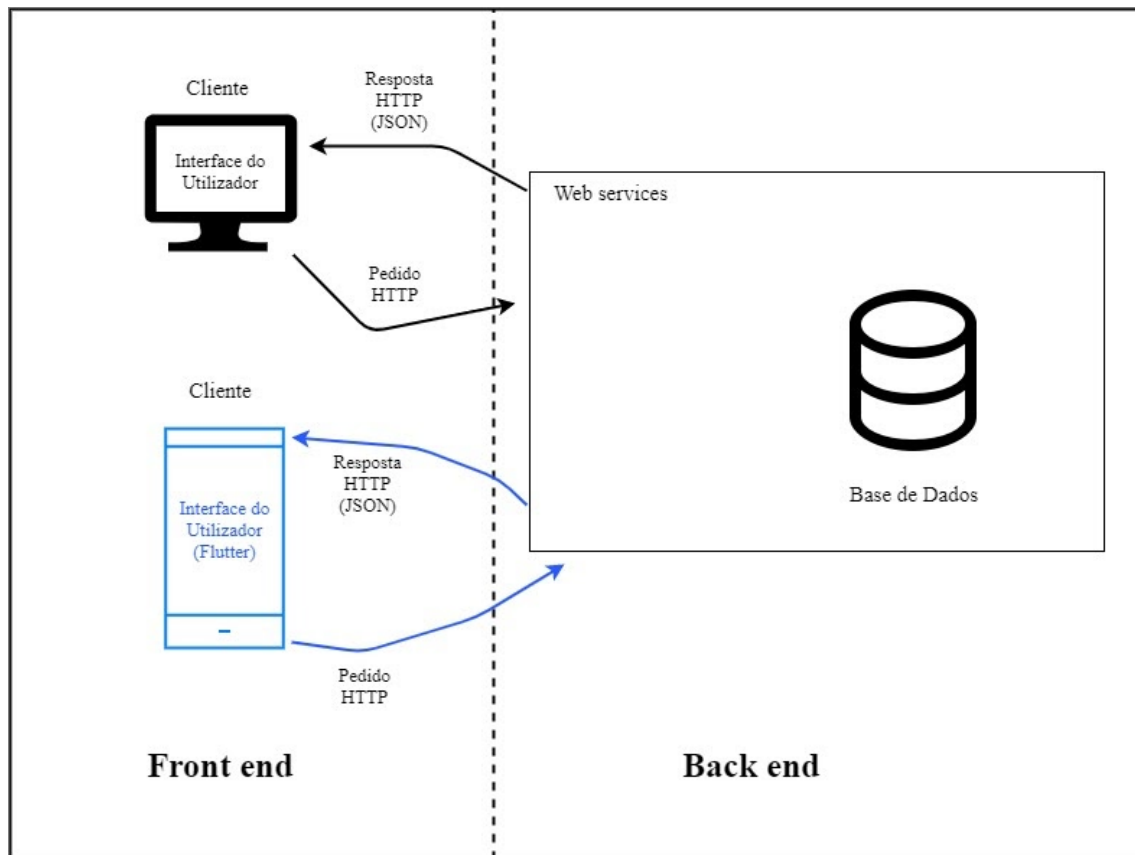


Figura 4 - Arquitetura do sistema geral

Tal como é visível na Figura 4, o sistema é constituído pelo *front end* responsável pela UI, desenvolvida com recurso ao Flutter, onde tanto membro como administrador interagem diretamente, e comunica com o *back end* via *web services* que, através do protocolo HTTP, transporta mensagens em formato JSON, de modo a consultar os dados presentes na mesma Base de Dados que a aplicação web usa, a fim de responder a todos os serviços requisitados. Trata-se de um Base de Dados MongoDB [12], previamente construída pelos alunos do TFC que serviu de premissa a este, baseada em documentos, ou seja, NoSQL, onde estão todas as informações relevantes como, por exemplo, a lista de eventos ou os membros registados na plataforma de gestão de eventos.

## 4.4 Protótipo Axure

Seguindo o fluxo natural de planeamento e de modo a complementar a solução já apresentada, surgiu a fase de prototipagem. Com recurso à ferramenta Axure RP 9 [13], o protótipo da aplicação móvel vai dar forma aos requisitos definidos, garantir a coerência da informação e simular o comportamento do sistema.

Para efeitos de demonstração, foram selecionados o ecrã de *login* (Figura 5), o ecrã principal (Figura 6), o menu do utilizador administrador (Figura 7), e o ecrã das estatísticas (Figura 8).

De realçar que todos os ecrãs desenvolvidos no Axure não têm como objetivo representar o estado final da aplicação mas sim de guiar na tomada de decisões UI/UX e durante o processo de escrita de código.

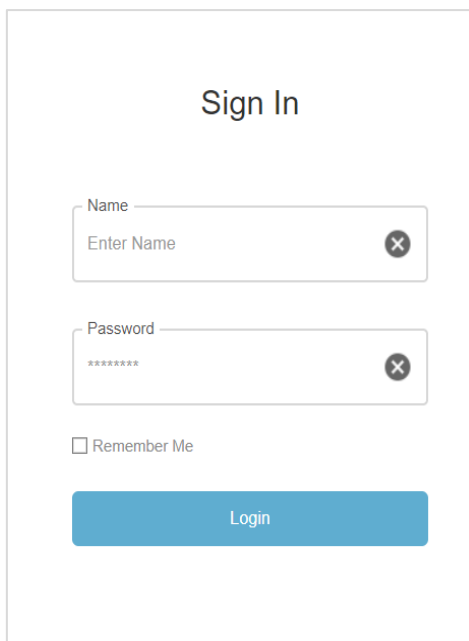


Figura 5 - Ecrã login Axure

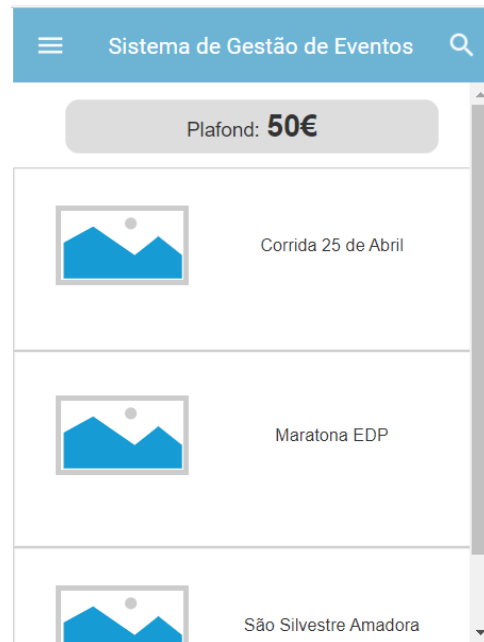


Figura 6 - Ecrã principal Axure

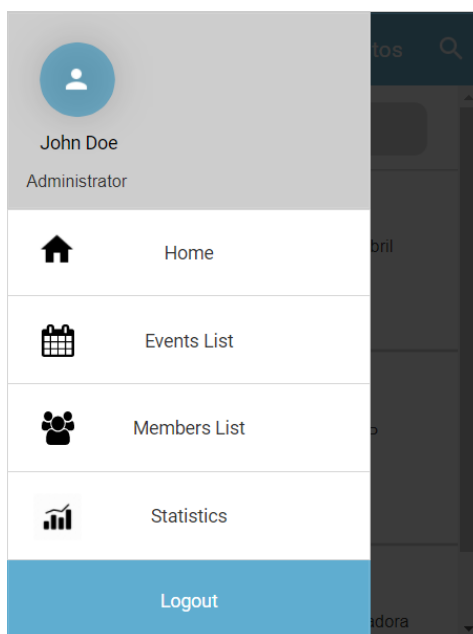


Figura 7 - Menu administrador Axure

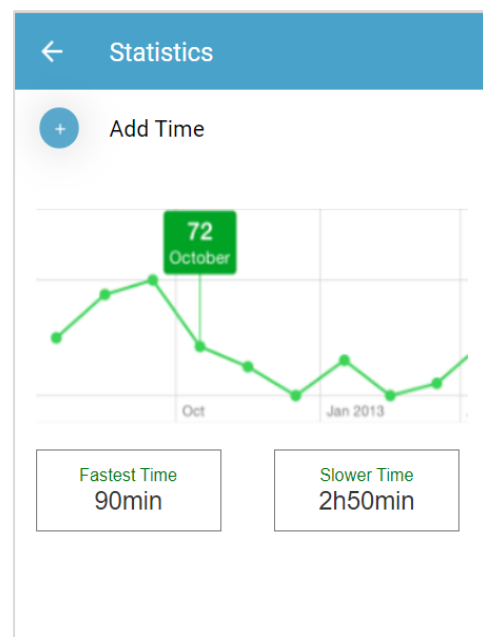


Figura 8 - Ecrã estatísticas Axure



## 4.5 Implementação

### 4.5.1 Responsive

Dada a construção deste protótipo, a aceitação dos requisitos apresentados no capítulo anterior pelas partes envolvidas e a configuração das tecnologias que a solução comporta, decidiu-se que estavam reunidas todas as condições para se proceder à fase de implementação de código.

O protótipo funcional foi desenvolvido através do Flutter, suportado pelo Android Studio, e concomitantemente testado através dos dispositivos móveis Pixel 2 API 30 na versão do Android 11.0 x86, Pixel API 23 na versão do Android 6.0 x86 e Nexus S API 23 na versão do Android 6.0 x86. Esta metodologia só foi possível recorrendo ao emulador integrado no IDE que garante quase todas as mesmas capacidades de um telemóvel real e permite simular a aplicação num leque de dispositivos com diferentes níveis de API de forma fácil e eficiente.

A escolha destes três dispositivos teve em conta fatores cruciais na preservação das características da aplicação:

- A versão do dispositivo - Analisar os diferentes comportamentos do projeto quando confrontado com uma API mais recente (R) *versus* uma API mais antiga (Marshmallow) é decisivo na escolha do tipo de ferramentas a aplicar, não limitando futuros utilizadores que tenham telemóveis mais antigos a não conseguir usufruir da aplicação no seu pleno potencial;
- O tamanho do ecrã - Procurar criar uma aplicação *responsive* e que com o mesmo código base se adapte às características de cada dispositivo, quer se esteja perante um ecrã de 5.0'' ou de 4.0'' deve ser tido constantemente em consideração. Este processo engloba uma estruturação do código de cada ecrã, recorrendo à classe `LayoutBuilder` [14] (constrói uma árvore de *widgets* dependente do tamanho do *widget* pai e quando esse tamanho é excedido, é ativado o `scroll`) e o método `MediaQuery.of()` [15] (retorna, por exemplo, as medidas do ecrã do dispositivo e adapta *widgets* consoante esse valor) como aparece nos excertos de código do Anexo 1.

### 4.5.2 Navegação

Além do estudo inicial realizado acerca da natureza *responsive* da aplicação, foi analisado como seria gerida a sua navegação. A escrita de código não se resume exclusivamente à execução do mesmo, é necessário haver uma estruturação e organização da forma como os ecrãs comunicam uns com os outros, transferem dados e, em repercussão, mudam de estado.

Nesta ótica, foram estudadas três alternativas:

- `setState()` [16]
- BLoC [17]
- Provider [18]

A exclusão do uso do `setState()` por si só foi imediata. Este padrão obedece à hierarquia, não permitindo obter uma referência direta a *widgets* que estejam noutras ramificações da árvore senão abaixo. Noutras palavras, se o estado de um *widget* mudasse num dado ecrã, não se verificaria a condição primordial de poder atualizar todos os ecrãs que partilhem este componente. Já o BLoC distingue-se pelo uso de *streams* e *sinks* no transporte de dados e atua como uma camada intermediária entre a UI e lógica de negócio (tornando o código mais acessível de perceber e manter) mas peca pela sua complexidade de implementação e código *boilerplate*.

A partir da informação recolhida, optou-se pela utilização do *package* Provider.

O Provider, tal como o BLoC, é entendido como um modelo que separa a lógica de negócio da UI e que tem a particularidade de reconstruir a UI sempre que ocorre uma mudança de estado.

Considere-se o *widget* responsável pelo *plafond* que estará presente em todos os ecrãs. Sempre que um utilizador inscreve-se numa prova é debitado o valor da atividade e o estado, partilhado por diferentes *widgets* e ecrãs, tem de ser atualizado. Mas como é que cada ecrã sabe que o valor do *plafond* mudou? Como é que esse valor é atualizado?

Através desta abordagem, os *widgets* não têm qualquer conceito do tipo de dados que têm de mostrar nem são responsáveis pela sua alteração. Em vez disso, existe uma classe chamada Plafond que estende ChangeNotifier e notifica as partes interessadas se ocorreu mudanças (uma forma de Observable) e assim proceder ao *rebuild* do *widget* Text como mostra o Anexo 2.

Seguindo o princípio de uma arquitetura de lógica de negócio, o projeto foi dividido em cinco pastas:

- Constants - todos os valores que se mantêm imutáveis ao longo da aplicação (por exemplo, as cores);
- Models - estrutura de dados encarregue de converter os pares chave/valor JSON às correspondentes variáveis da entidade;
- Providers - contém o modelo Provider para cada *view* dos *widgets*;
- Screens - lida somente com a UI, isto é, como os dados são apresentados e como é realizada a receção de eventos espotelados pelo utilizador (cliques num dado botão, *scrolls*);
- Services - dá a resposta do servidor quando se faz chamadas à Base de Dados;

À semelhança da arquitetura Model View ViewModel (MVVM), ilustrada na Figura 9, os ficheiros presentes na pasta Providers e Models replicam o mesmo comportamento do View Model, ou seja, estabelecer a interação entre os ecrãs e o resultados dos pedidos dos *web services* à Base de Dados, fornecendo os dados de forma a que a UI os possa apresentar, mesmo não sabendo nada sobre a mesma.

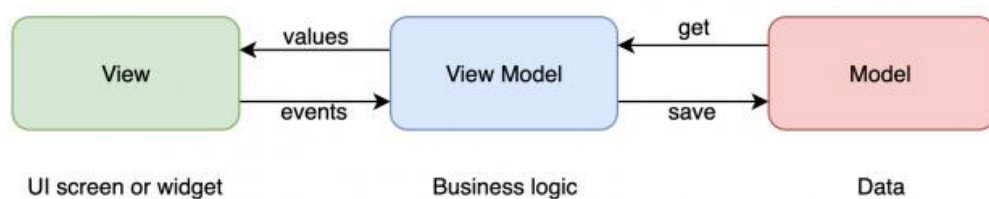


Figura 9 - Arquitetura MVVM

Como previamente salientado, este padrão é pertinente na medida em que promove uma maior disciplina durante a programação e, com a sua estratégia baseada na segregação de componentes, oferece uma cómoda oportunidade de crescimento da aplicação e de testagem de cada classe independente de outrem, o que traduz num benefício a médio/longo prazo se se proceder à continuidade do trabalho.

## 4.6 Protótipo Flutter

Do culminar da sequência de fases executadas e discutidas previamente, resultou um protótipo funcional capaz de responder às carências que condicionaram o surgimento desta proposta de TFC e que satisfaz todos os requisitos validados.

Tal como é visível na Figura 10, o protótipo começa com um *splash screen*, um ecrã inicial que introduz o utilizador à aplicação e responsável pela primeira perceção. A escolha do laranja como a cor predominante está interligada com o facto do logotipo da empresa ser composto por este tom.



Figura 10 - Splash screen

Do *splash screen*, surge o ecrã do *login* (Figura 11), que irá direccionar o utilizador para o ecrã principal da aplicação (Figura 12) se se verificarem as credenciais introduzidas nos devidos campos. Esta ecrã é comum aos dois tipos de utilizadores, com a particularidade de quando confrontados com o ícone do canto superior esquerdo, para o administrador é revelado o menu da Figura 13 e para o membro o menu da Figura 14.

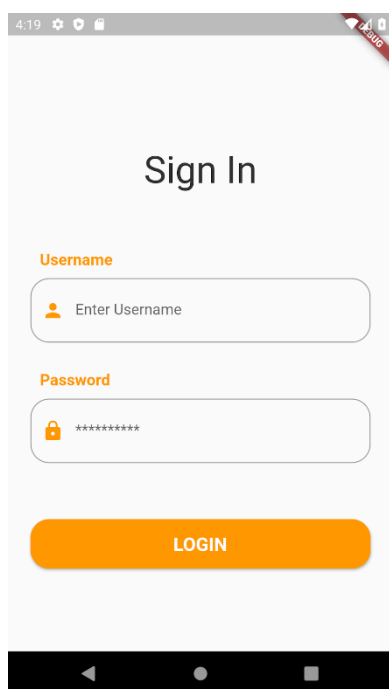


Figura 11 - Ecrã login

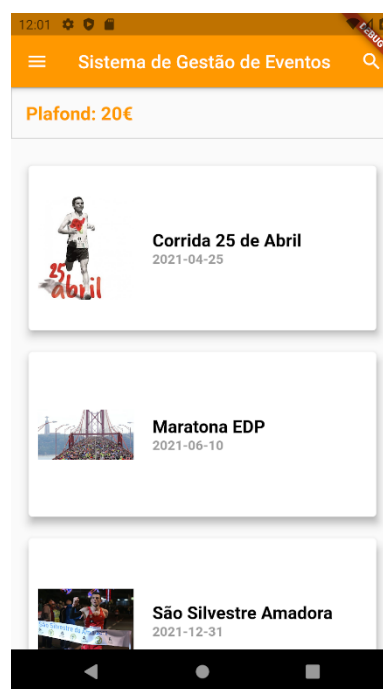


Figura 12 - Ecrã principal

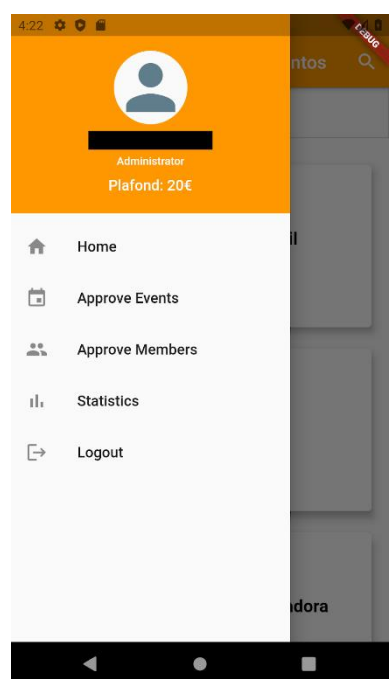


Figura 13 - Menu administrador

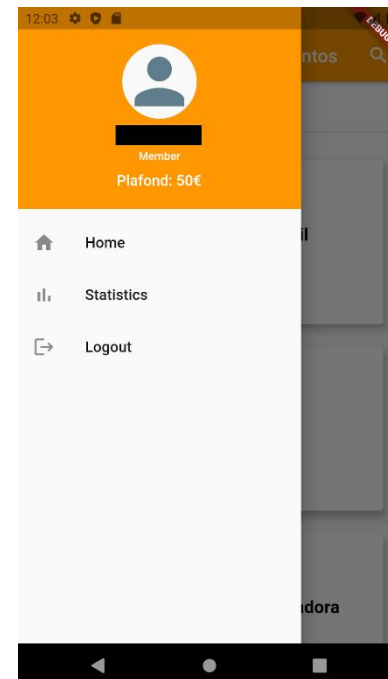


Figura 14 - Menu membro

Ainda no ecrã principal, salienta-se que ambos os utilizadores, ao clicar no ícone presente no canto superior direito, têm disponível um campo de pesquisa que, ao introduzir o nome de um evento que desejem, o mesmo será apresentado, facilitando o processo de procura (Figura 15). É fundamental esclarecer que todas as provas desportivas que constituem o protótipo são de teor *dummy data*, isto é, dados fictícios, uma vez que com a atual crise pandémica não há organização de eventos desportivos abertos ao público.

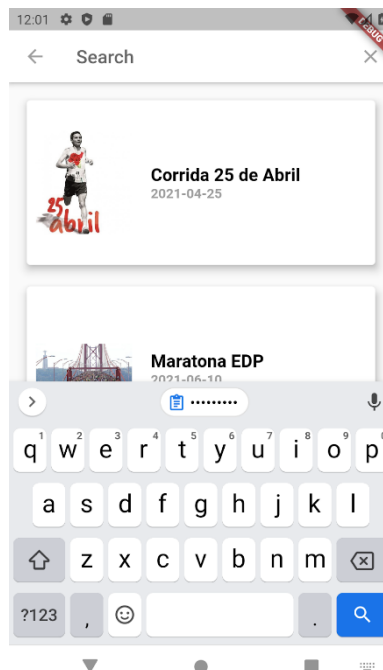


Figura 15 - Campo de pesquisa

Ao selecionar um dado evento, neste caso, o primeiro, os utilizadores serão encaminhados para o ecrã da Figura 16 e podem proceder à sua inscrição.

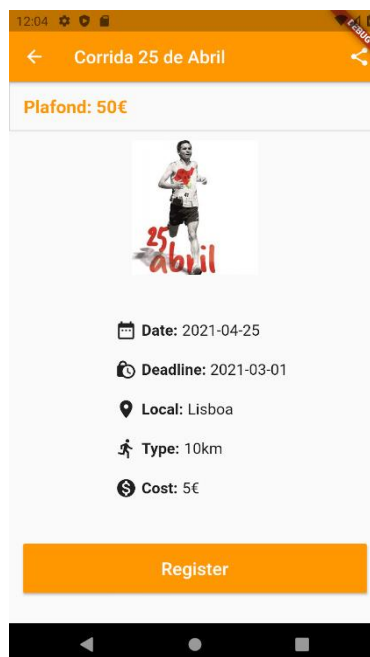


Figura 16 - Ecrã detalhe de evento

Para se inscreverem, basta clicar no botão “Register” e o utilizador será registado na atividade desportiva se, e só se, o *plafond* for igual ou superior ao custo da mesma e o prazo de inscrição ainda não tiver terminado. Se estas condições se verificarem, o botão “Register” atualiza para um botão “Cancel” e o utilizador em causa pode cancelar a sua inscrição a qualquer momento, sendo novamente creditado o valor da prova. Este processo está ilustrado na Figura 17.

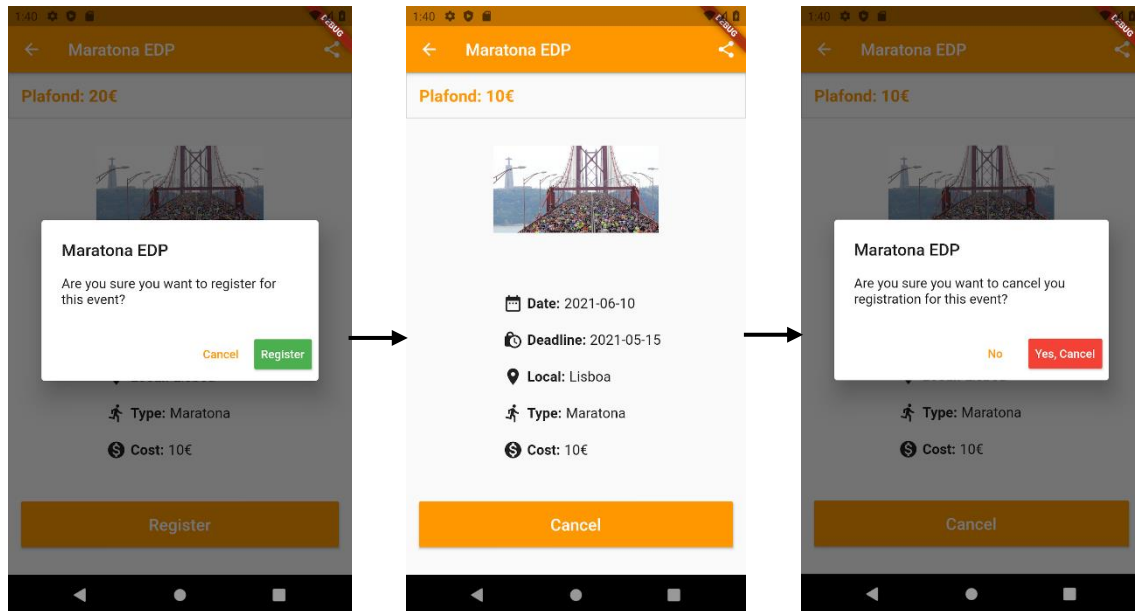


Figura 17 - Inscrição e cancelamento de evento

Por outro lado, se a data limite de inscrição já tiver terminado aparece a mensagem presente na Figura 18.

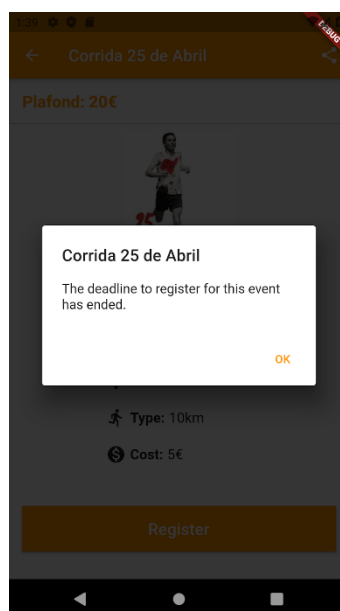


Figura 18 - Mensagem de erro prazo

Para um *plafond* inferior ao exigido pela atividade, surge a mensagem da Figura 19.

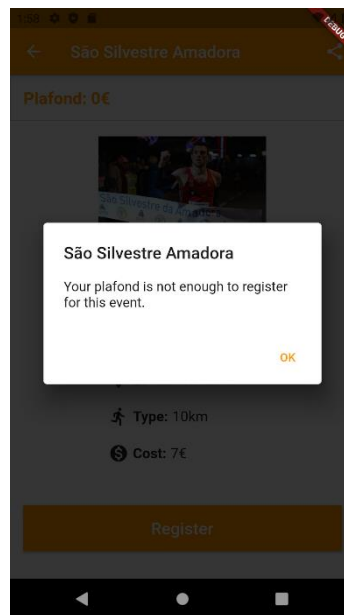


Figura 19 - Mensagem de erro plafond

Adicionalmente, cada ecrã com o detalhe do evento oferece a opção de partilhar a prova nas redes sociais ao interagir com o ícone no canto superior direito.

Enquanto que os ecrãs acima são comuns aos dois tipos de utilizador, a Figura 20 e a Figura 21 incorpora as componentes de aprovação ou rejeição de eventos e membros, respetivamente, exclusivas do administrador.

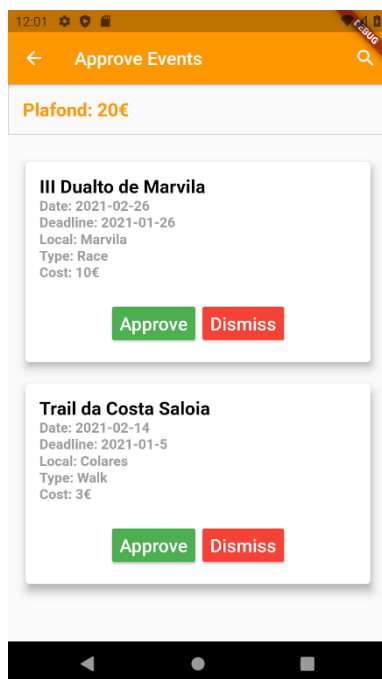


Figura 20 - Ecrã aprovação de eventos

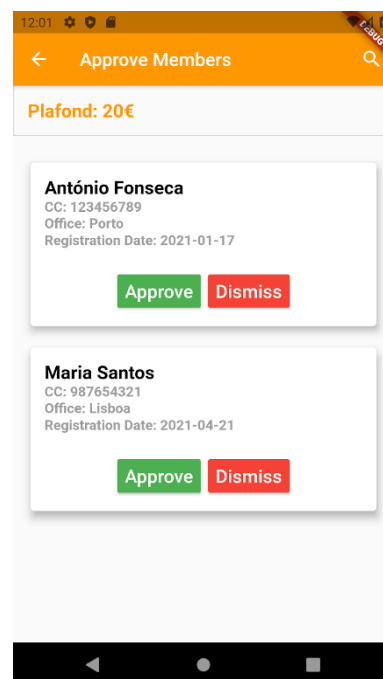


Figura 21 - Ecrã aprovação de membros

O administrador tem o poder de aprovar o evento/membro ao clicar no botão “Approve”, de rejeitar ao clicar no botão “Dismiss” e de efetuar a pesquisa por nome. Quando um evento é aprovado, o mesmo é retirado desta lista e posteriormente adicionado ao *home* juntamente com os outros eventos. Para o membro, o processo é semelhante e este terá acesso à aplicação. O ato de rejeição, tanto para evento como para membro, resulta na sua automaticamente remoção das respetivas listas. Como se tratam de ações sensíveis, a aplicação pergunta sempre se o administrador tem a certeza que pretende proceder (Figura 22).

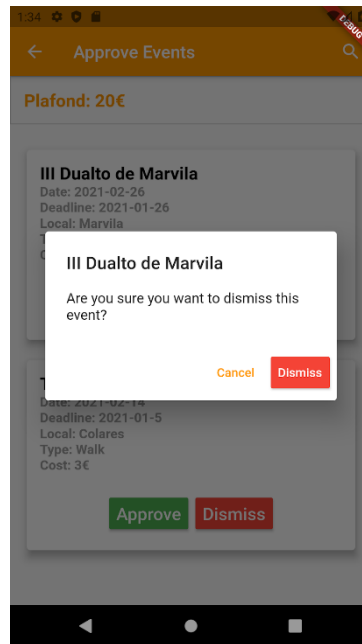


Figura 22 - Rejeitar evento

Por fim, tem-se na Figura o ecrã cujo seu teor baseia-se numa relação simbiótica entre o tempo introduzido e o gráfico produzido - as estatísticas.

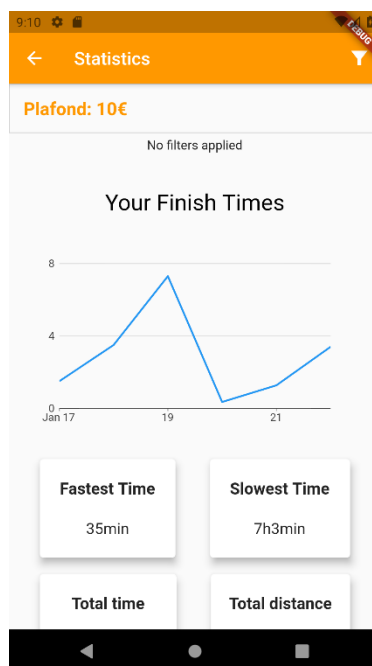
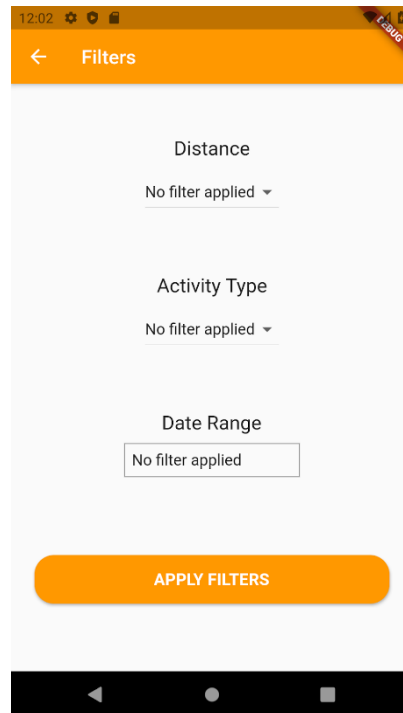


Figura 23 - Ecrã estatísticas



Neste ecrã, os utilizadores podem consultar um gráfico onde o eixo do x representa a data em que foi realizada a prova e o eixo do y o tempo, em horas, que demorou. Logo a seguir, surgem quatro *cards* correspondentes ao tempo mais rápido, ao tempo mais lento, ao total de km percorridos e o total de tempos. Observa-se que no cabeçalho do ecrã há uma indicação visual de “No filters applied”, ou seja, por omissão, todos os valores inicialmente derivam do conjunto global de tempos inseridos.

Se o utilizador pretender filtrar os resultados, basta clicar no ícone no canto superior direito e selecionar o filtro que tenciona (Figura 24).



**Figura 24 - Ecrã filtros**

Estão disponíveis três filtros: distância percorrida (em km), tipo de atividade (corrida ou caminhada) e intervalo de datas (suportado por um calendário da biblioteca `date_range_picker`

[19] ). A título de exemplo, quando aplicados o filtro de distância de 10km e tipo de atividade corrida obtém-se os resultados da Figura 25.

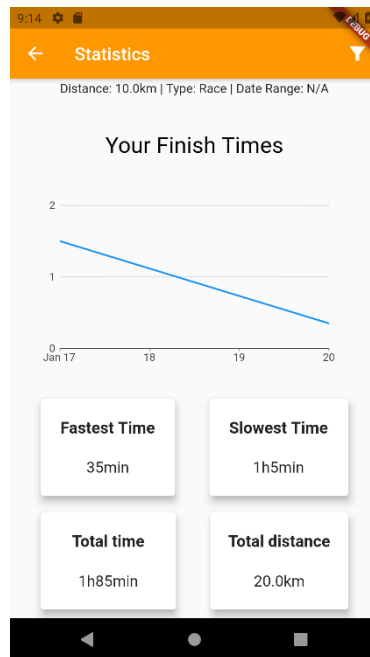


Figura 25 - Ecrã estatísticas após aplicação de filtros

Todas estas variáveis podem ser inseridas quando o botão “Add Time”, que se encontra no fim do ecrã das estatísticas, é pressionado. Para um tempo ser registado com sucesso, a distância percorrida e o tempo de conclusão têm de ser valores positivos e não nulos (Figura 26).

Add a Time

Activity Type:

☒ Race ☐ Walk

Date: 2021-04-22

Distance: km

Finish Time: 0:0:0 h

Enter a valid Distance and Finish Time value

ADD

+ Add a Time

Figura 26 - Registo de tempos

## **4.7 Código Fonte**

O código fonte está disponível para consulta no seguinte repositório Git: [https://github.com/catarina-21805632/projeto\\_tfc.git](https://github.com/catarina-21805632/projeto_tfc.git).

## **4.8 Vídeo**

Todo funcionamento do protótipo funcional aqui descrito é posto em prática no seguinte link de Youtube: <https://youtu.be/o1l1hC7HLdo>.

## 5 Benchmarking

Para ganhar conhecimento do mercado, identificar pontos críticos que promovam o sucesso do projeto e arranjar metodologias que tornem a aplicação móvel capaz de responder aos serviços que as aplicações concorrentes já oferecem e saber o que fazer para inovar e preencher as lacunas que possam haver, é de extrema importância realizar o processo de análise e pesquisa de aplicações que sirvam o mesmo propósito que a deste trabalho.

Em comparação com o que já existe no que toca a plataformas de gestão de eventos desportivos temos:

- Eventsport [20]
- Running Lisboa [21]
- Pronto a Correr [22]
- Run Races [23]
- RamRun [24]

### Eventsport

#### *Descrição Oficial*

“Aplicação pessoal para participação em eventos desportivos da Eventsport. Organização de eventos desportivos, gestão de inscrições, controlo de tempos, gestão de corridas e maratonas, equipas e atletas.”

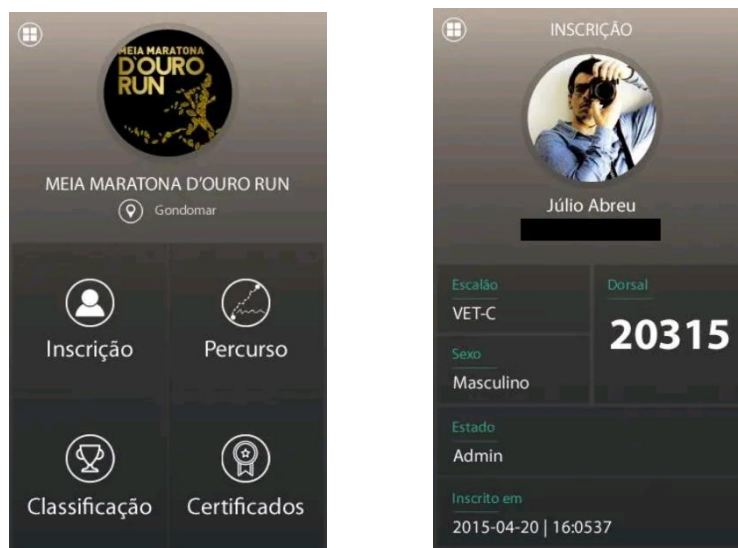


Figura 27 - Aplicação Eventsport

## Running Lisboa

### Descrição Oficial

“Aplicativo oficial do Maratona Clube de Portugal em Lisboa, desenvolvido por MYLAPS.”

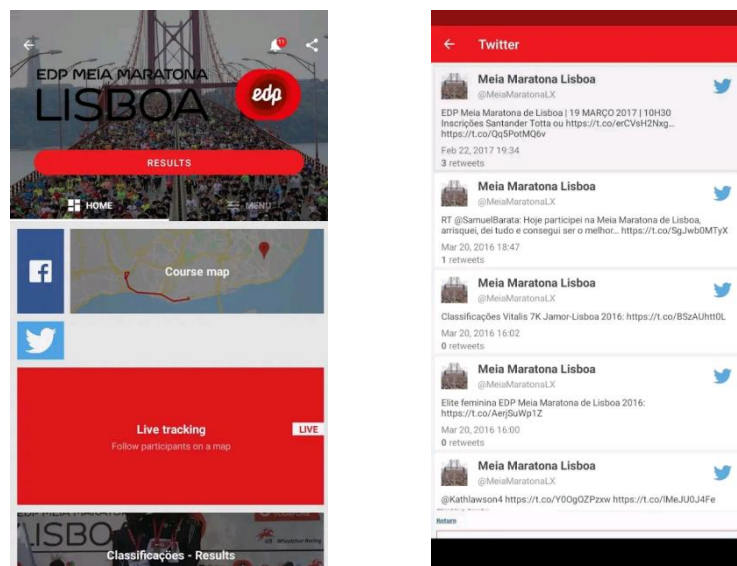


Figura 28 - Aplicação Running Lisboa

## Pronto a Correr

### Descrição Oficial

“Calendário de provas de estrada e de trail running.”

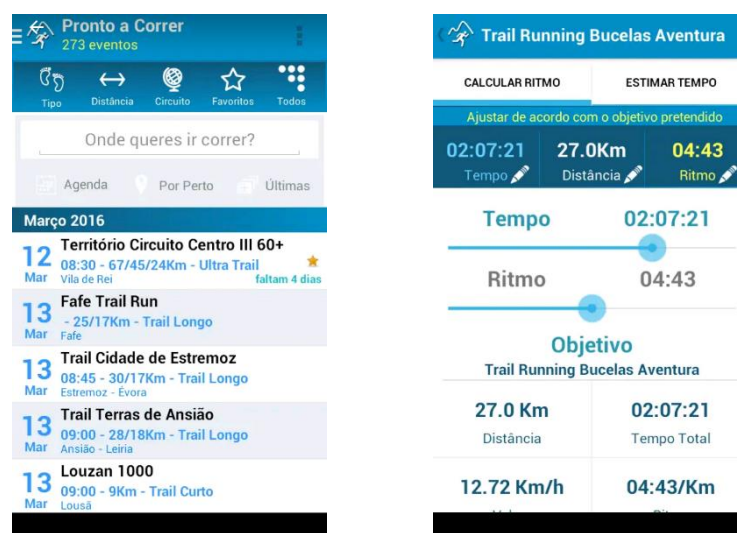


Figura 29 - Aplicação Pronto a Correr

## Run Races

### Descrição Oficial

“Encontre a sua próxima corrida em execução.”

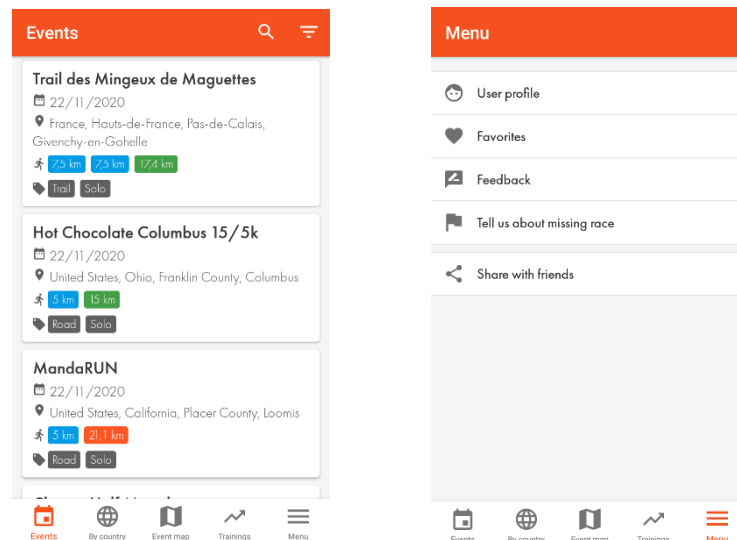


Figura 30 - Aplicação Run Races

## RamRun

### Descrição Oficial

“Calendário de eventos em execução (marchas, corridas, duatlo...) em todos os Estados Unidos.”

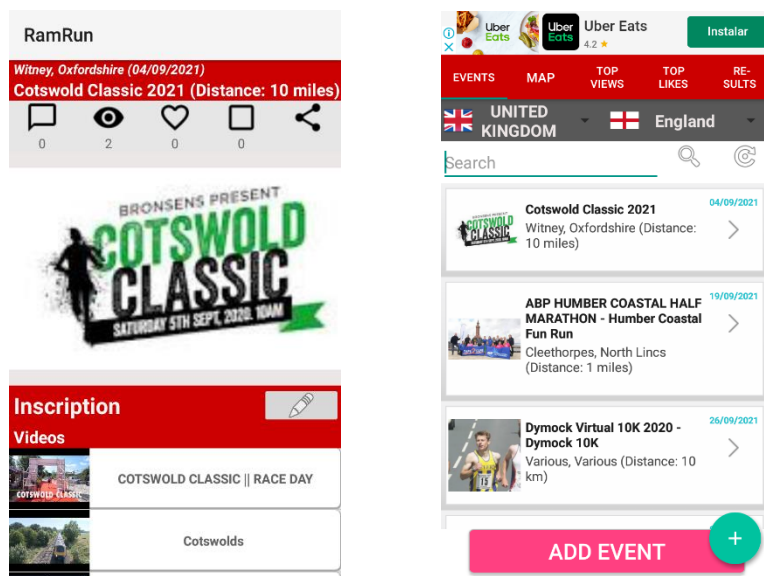


Figura 31 - Aplicação RamRun

Como referido, as aplicações acima definidas têm por base o mesmo propósito: os utilizadores inscrevem-se em eventos e os administradores realizarem a gestão dos mesmos. Estes sistemas concorrentes distinguem-se, nomeadamente, pelo elevado número de *downloads*, pela sua comunidade e utilizadores ativos e pelas excelentes classificações e *reviews*, que podem ser um indicador de popularidade e grau de satisfação, presentes em serviços de distribuição de aplicações móveis como o Google Play Store e/ou o serviço App Store da Apple.

De modo a proceder à comparação destas aplicações com a aplicação proposta a desenvolver foi construída a Tabela 16:

**Tabela 16 - Comparação da aplicação a ser desenvolvida com aplicações concorrentes**

Nome	Rating	Plataformas	Preço	Registrar tempo (S/N)	Estatísticas (S/N)	Redes Sociais integradas (S/N)
Sistema de gestão de eventos	N/A	Android e iOS	Grátis	S	S	S
Eventsport	4,0	Android e iOS	Grátis	S	N	N
Running Lisboa	N/A	Android e iOS	Grátis	N	N	S
Pronto a Correr	4,5	Android	Grátis	S	S	N
Run Races	4,8	Android e iOS	Grátis	N	N	S
RamRun	3,8	Android	0,99€ - 3,09€	N	N	S

Por inspeção direta da Tabela 16, pode-se retirar que todas as aplicações móveis em objeto de análise apresentam algumas das características em estudo mas somente a solução proposta ao longo deste relatório inova neste sentido e incorpora todos os atributos considerados. É relevante mencionar que, tanto a Pronto a Correr como a RamRun só se encontram disponíveis para dispositivos Android. Com base na ideia explorada no capítulo anterior, esta característica é interpretada como uma desvantagem em relação às restantes aplicações que possuem caráter multiplataforma, sendo que, apesar do Android ter uma maior quota no mercado, o iOS representa uma parte bastante significativa dos utilizadores de *smartphones*.

Juntamente, constata-se que todas as aplicações com exceção da RamRun são grátis. Esta última difere-se no facto de ter anúncios que, ao pagar uma certa quantia, são omitidos.

## 6 Plano de testes e validação

A construção de um plano de testes fidedigno que empregue metodologias com base na qualidade, que escolha as condições adequadas em que serão realizados e que faça uso de critérios de avaliação transparentes são fundamentais para a validação prática e operacional da solução desenvolvida ao longo dos capítulos abordados anteriormente.

Erroneamente entende-se por testes como um meio exclusivamente utilizado para identificar vulnerabilidades de um *software*. Muito mais que demonstrar o bom ou mau funcionamento de um produto, é tirar ilações quanto ao cumprimento dos objetivos solicitados na fase de levantamento de requisitos, garantindo que todas as especificações vão de encontro ao esperado pelo cliente.

Como critérios de teste deverá ser assegurado que a aplicação móvel sob teste satisfaça as diretrizes dos requisitos funcionais e não funcionais, vai de encontro às especificações de qualidade definidas pelos intervenientes e que qualquer *bug* identificado seja resolvido antes da entrega final. Deve ser testado cada componente individualmente, até se conseguir o desempenho global do sistema.

Outro elemento importante no contexto deste TFC será a realização de um questionário em forma de teste de usabilidade [25] com a participação de terceiros que beneficiem diretamente da aplicação.

A calendarização destes testes encontra-se disponível para consulta na Figura 32.

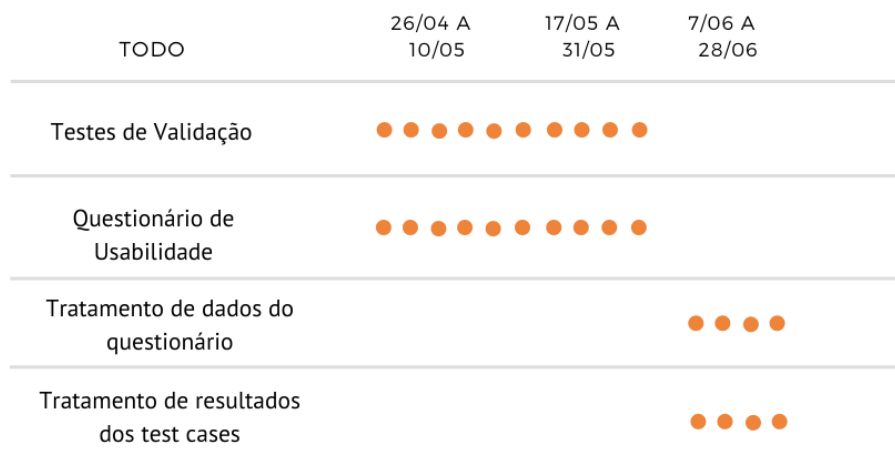


Figura 32 - Calendário de testes



## 7 Calendário

Em conformidade com a solução pretendida e de maneira a que seja cumprido e executado todos os pontos dentro dos prazos estabelecidos, é proposto o plano de trabalho e cronograma apresentado na Figura 33:

	23/10/2020	27/10/2020	27/11/2020	27/12/2020	22/01/2021	22/02/2021	22/03/2021	23/04/2021	23/05/2021	25/06/2021
Instalação e configuração do Flutter no ambiente de trabalho										
Estudar a ferramenta Flutter e como aceder a web services a partir desta										
Levantamento e análise detalhada de requisitos										
Instalação e execução do sistema na sua forma web										
Implementação da aplicação móvel em código										
Implementação de testes de validação										
Implementação de um protótipo funcional										
Versão final do trabalho										

**Figura 33 - Gráfico de Gant do plano de trabalho proposto**



## Bibliografia

- [1] Free online meeting scheduling tool | Doodle. [Online]. Available: <https://doodle.com/en/>. [Acesso: 22-10-2020].
- [2] What is Agile Software Development?, *Visual Paradigm* [Online]. Available: <https://www.visual-paradigm.com/scrum/what-is-agile-software-development/>. [Acesso: 25-01-2021].
- [3] What is User Story?, *Visual Paradigm*. [Online]. Available: <https://www.visual-paradigm.com/guide/agile-software-development/what-is-user-story/>. [Acesso: 01-02-2021].
- [4] FAQ, *Flutter*. [Online]. Available: <https://flutter.dev/docs/resources/faq#what-devices-and-os-versions-does-flutter-run-on>. [Acesso: 24-10-2020].
- [5] J. Nielsen and R. Budi, *Mobile Usability*. New Riders Press, 2012.
- [6] Flutter - Beautiful native apps in record time. [Online]. Available: <https://flutter.dev/>. [Acesso: 24-10-2020].
- [7] Statcounter. (2020, Setembro). *Mobile Operating System Market Share Worldwide*. [Online]. Available: <https://gs.statcounter.com/os-market-share/mobile/worldwide>. [Acesso: 08-11-2020].
- [8] Dart programming language | Dart. [Online]. Available: <https://dart.dev/>. [Acesso: 24-10-2020].
- [9] Skia in Flutter & Fuchsia. [Online]. Available: <https://skia.org/dev/flutter>. [Acesso: 24-10-2020].
- [10] JSON and serialization. [Online]. Available: <https://flutter.dev/docs/development/data-and-backend/json>. [Acesso: 24-10-2020].
- [11] Welcome to Common-Lisp.net!. [Online]. Available: <https://common-lisp.net/>. [Acesso: 14-11-2020].
- [12] The most popular database for modern apps | MongoDB. [Online]. Available: <https://www.mongodb.com/>. [Acesso: 14-11-2020].
- [13] Axure RP 9 - Prototypes, Specifications, and Diagrams in One Tool, *Axure*. [Online]. Available: <https://www.axure.com/>. [Acesso: 19-03-2021].
- [14] LayoutBuilder class - widgets library - Dart API, *Api.flutter.dev*. [Online]. Available: <https://api.flutter.dev/flutter/widgets/LayoutBuilder-class.html>. [Acesso: 01-04-2021].
- [15] of method - MediaQuery class - widgets library - Dart API, *Api.flutter.dev*. [Online]. Available: <https://api.flutter.dev/flutter/widgets/MediaQuery/of.html>. [Acesso: 01-04-2021].
- [16] setState method - State class - widgets library - Dart API, *Api.flutter.dev*. [Online]. Available: <https://api.flutter.dev/flutter/widgets/State/setState.html>. [Acesso: 01-04-2021].
- [17] Bloc State Management Library, *BlocLibrary.dev*. [Online]. Available: <https://bloclibrary.dev/#/>. [Acesso: 01-04-2021].
- [18] provider | Flutter Package, *Dart packages*. [Online]. Available: <https://pub.dev/packages/provider>. [Acesso: 01-04-2021].
- [19] date\_range\_picker | Flutter Package, *Dart packages*. [Online]. Available: [https://pub.dev/packages/date\\_range\\_picker](https://pub.dev/packages/date_range_picker). [Acesso: 01-04-2021].
- [20] Admeus. (2016). *Eventsport* (Version 0.0.2) [Mobile App]. Available: <https://play.google.com/store/apps/details?id=pt.eventsport.app2&hl=pt&gl=US>
- [21] MYLAPS Experience Lab. (2020). *Running Lisboa* (Version 1.0) [Mobile App]. Available: [https://play.google.com/store/apps/details?id=com.mylaps.eventapp.runninglisboa&hl=en\\_IN](https://play.google.com/store/apps/details?id=com.mylaps.eventapp.runninglisboa&hl=en_IN)
- [22] Outside Dreamz. (2014). *Pronto a Correr* (Version 1.5.3) [Mobile App]. Available: [https://play.google.com/store/apps/details?id=com.outsidedream.myrun&hl=pt\\_PT](https://play.google.com/store/apps/details?id=com.outsidedream.myrun&hl=pt_PT)
- [23] Mobilife. (2017). *Run Races* (Version 1.11.0) [Mobile App]. Available: [https://play.google.com/store/apps/details?id=org.runraces&hl=pt\\_PT&gl=US](https://play.google.com/store/apps/details?id=org.runraces&hl=pt_PT&gl=US)
- [24] RamSoft. (2019). *RamRun* (Version 1.35) [Mobile App]. Available: [https://play.google.com/store/apps/details?id=com.events.running&hl=pt\\_BR](https://play.google.com/store/apps/details?id=com.events.running&hl=pt_BR)

[25] K. Moran, "Usability Testing 101", *Nielsen Norman Group*, 2019. [Online]. Available: <https://www.nngroup.com/articles/usability-testing-101/>.



## Anexo 1 - Implementação do LayoutBuilder() e MediaQuery.of()

```
body: new LayoutBuilder(  
  builder: (BuildContext context, BoxConstraints  
viewportConstraints){  
    return SingleChildScrollView(  
      child: ConstrainedBox(  
        constraints: BoxConstraints(  
          minHeight: viewportConstraints.maxHeight,  
        ),  
      ),  
    ),  
  );  
);
```

```
TextSpan(  
  text: '\n${event.date}',  
  style: TextStyle(  
    color: Colors.grey,  
    fontSize: (15/720)* MediaQuery.of(context).size.height  
  ),  
),
```

## Anexo 2 - Implementação do Provider

```
class Plafond with ChangeNotifier{

  int _plafond = 0;

  int get value => _plafond;

  setPlafond(int value) {
    _plafond = value;
    notifyListeners();
  }

  void incrementPlafond(int value) {
    _plafond += value;
    notifyListeners();
  }

  void decrementPlafond(int value){
    _plafond -= value;
    notifyListeners();
  }
}
```

```
child: Consumer<Plafond>(
  builder: (context, plafond, child) => Text(
    'Plafond: ${plafond.value}€',
    style: TextStyle(fontSize: 20.0, color: Colors.orange, fontWeight:
    FontWeight.bold),
  ),
),
```

## Glossário

LEI	Licenciatura em Engenharia Informática
TFC	Trabalho Final de Curso
PME	Pequena e Média Empresa
SO	Sistemas Operativos
UX	Experiência de Utilizador
UI	Interface de Utilizador
ARM	Advanced RISC Machine
REST	Representational State Transfer
JSON	JavaScript Object Notation
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
API	Interface de Programação de Aplicações
CPU	Unidade Central de Processamento
BLoC	Business Logic of Component
MVVM	Model View ViewModel