IPCA-EST-MEI-PCC

<Insert Project Name>

Software Architecture Document (SAD)

CONTENT OWNER

<owner name>

|  |  |  |
| --- | --- | --- |
| DOCUMENT NUMBER: | RELEASE/REVISION: | RELEASE/REVISION DATE: |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| **BACKGROUND (Delete this page)**  This template is based on the Software Engineering Institute’s “View and Beyond” method for documenting software architectures, as described in Clements, et al., [*Documenting Software Architecture: Views and Beyond*](http://www.sei.cmu.edu/architecture/books.html) (Addison Wesley, 2002), as well as in *Software Architecture for Developers* (Brown, 2019).  **TIPS FOR USING THIS TEMPLATE**  To create an instance of this document:  Insert relevant information on cover sheet and in placeholders throughout.  Insert relevant information in page header: Move to a page of the body of the report, select *View > Header and Footer* from the main menu, and then replace relevant information in the header box at the top of the page.  To update the contents and page numbers in the Table of Contents, List of Figures, and List of Tables:  Position the cursor anywhere in the table to be updated.  Click the *F9* function key.  Answer “Update entire table”.  To insert a figure or table caption:  From the main menu, choose *Insert > Reference > Caption* and then either *Figure* or *Table* as needed.  Click the OK button.  Add a colon and a tab stop after the figure number in the caption itself.  The caption should use the *Caption* style.  Add a colon and a tab stop after the table/figure number in the caption itself.  **TIPS FOR MAKING YOUR DOCUMENT MORE READABLE**  A gray box containing *CONTENTS OF THIS SECTION* is provided at the beginning of most sections and subsections. After determining what specific information will be included in your document, you can remove this gray box or leave it to serve as a quick-reference section overview for your readers. In the case that text has been provided in the template, inspect it for relevance and revised as necessary.  Consider hyperlinking key words used in the document with their entries in the [Glossary](#_Glossary) or other location in which they are defined. Choose *Insert > Hyperlink*.  Don’t leave blank sections in the document. Mark them “To be determined” (ideally with a promise of a date or release number by which the information will be provided) or “Not applicable.”  Consider packaging your SAD as a multi-volume set of documentation. It is often helpful to break your documentation into more than one volume so that the document does not become unwieldy. There are many ways that this can be accomplished. The structuring of the document must support the needs of the intended audience and must be determined in the context of the project. Each document that you produce should include the date of issue and status; draft, baseline, version number, name of issuing organization; change history; and a summary. A few decomposition options are:  *A 2-Volume approach:* Separate the documentation into two volumes; one that contains the views of the software architecture and one that contains everything else. A common variant of this approach has one volume per view, and one volume for everything else.  *A 3-Volume approach:* Document organizational policies, procedures, and the directory in one volume, system specific overview material in a second, and view documentation in a third.  *A 4-Volume approach:* Create one volume for each viewtype (Brown, 2019) (module, component-and-connector, allocation) that contains the documentation for the relevant [views](#view). Include all of the other information in the fourth volume.  Software interfaces are often documented in a separate volume.  In *any* case, the information should be arranged so that readers begin with the volume containing the Documentation Roadmap (Section 1 in this template). |

Table of Contents

[1 Introduction and Goals 3](#_Toc125153349)

[1.1Requirements Overview 3](#_Toc125153350)

[1.2Quality goals 3](#_Toc125153351)

[1.3Stakeholders 3](#_Toc125153352)

[2 Constraints 5](#_Toc125153353)

[2.1Technical Constraints 5](#_Toc125153354)

[2.2Organizational Constraints 5](#_Toc125153355)

[2.3Conventions 5](#_Toc125153356)

[3 Context 6](#_Toc125153357)

[3.1Business Context 6](#_Toc125153358)

[3.2Deployment Context 6](#_Toc125153359)

[4 Solution strategy 7](#_Toc125153360)

[4.1Introduction to the Strategy 7](#_Toc125153361)

[4.2Structure of the system 7](#_Toc125153362)

[4.3System Strategy 7](#_Toc125153363)

[4.4The Connection of the System 7](#_Toc125153364)

[5 Building Block View 8](#_Toc125153365)

[5.1Level 1 8](#_Toc125153366)

[5.2Level 2 8](#_Toc125153367)

[6 Runtime View 9](#_Toc125153368)

[6.1“Feature X” 9](#_Toc125153369)

[7 Deployment View 10](#_Toc125153370)

[7.1Operating System Infrastructure (Windows, IOS, others) 10](#_Toc125153371)

[8 Technical and crosscutting concepts 11](#_Toc125153372)

[8.1Dependencies Between Modules 11](#_Toc125153373)

[8.2Domain Model 11](#_Toc125153374)

[8.3User Interface 11](#_Toc125153375)

[8.4Exception and Error Handling 11](#_Toc125153376)

[8.5Logging and Tracing 11](#_Toc125153377)

[8.6Testability 11](#_Toc125153378)

[9 Design Decisions 12](#_Toc125153379)

[9.1How Does the Engine Communicate with the Outside World? 12](#_Toc125153380)

[9.2Are Position Objects Changeable or Not? 12](#_Toc125153381)

[10 Quality Requirements 13](#_Toc125153382)

[10.1Utility Tree 13](#_Toc125153383)

[10.2Quality Scenarios 13](#_Toc125153384)

[11 Glossary 14](#_Toc125153385)

[11.1Terms 14](#_Toc125153386)

[12 Source Code 15](#_Toc125153387)

[13 Sample Figures & Tables 16](#_Toc125153388)

List of Figures

[Figure 1: Sample Figure 22](#_Toc51125206)

List of Tables

[Table 1 - Quality Goals 3](#_Toc125153420)

[Table 2 - Stackeholders 4](#_Toc125153421)

[Table 3 - Technical constraints 5](#_Toc125153422)

[Table 4 - Organizational constraints 5](#_Toc125153423)

[Table 5 - Conventions 5](#_Toc125153424)

[Table 6 - Quality goals and Architectures 7](#_Toc125153425)

[Table 7 - Logical components cescription – Level 1 8](#_Toc125153426)

[Table 8 - Logical sub-components description – Level 2 8](#_Toc125153427)

[Table 9 - Quality Scenarios 13](#_Toc125153428)

[Table 10 – Glossary terms 14](#_Toc125153429)

[Table 11: Sample Table 16](#_Toc125153430)

# Introduction and Goals

Atualmente, em Portugal a obesidade e o excesso de peso é um problema que tem vindo a aumentar com o decorrer dos anos. Estima-se que cerca de 50% da população portuguesa tenha excesso de peso ou obesidade e que 30% das crianças tenham o mesmo problema. O objetivo da nossa aplicação é promover o exercício físico fornecendo uma plataforma que facilita o agendamento de uma atividade que a maioria dos portugueses gosta de fazer: jogar futebol.

## Requirements Overview

Antes de iniciar o desenvolvimento de uma aplicação, é importante definir claramente o que se espera da aplicação e como ela deve funcionar. Isso inclui considerações sobre o público-alvo, as funcionalidades desejadas, as restrições técnicas e os requisitos de desempenho.

Nesta secção, discutiremos os principais requisitos da aplicação.

### Requisitos funcionais

Os requisitos funcionais são uma parte fundamental do processo de desenvolvimento de software. São utilizados para definir as funcionalidades que a aplicação deve ter e servem como uma ponte entre as expectativas dos utilizadores e a implementação técnica.

Para a Kickoff foram definidos os seguintes requisitos funcionais:

* RF01 - Serviço de autenticação – A aplicação deve possuir um serviço de autenticação com registo, login e recuperação de palavra-passe.
* RF02 – Criar equipa – A aplicação deve possuir a funcionalidade de criação de equipas.
* RF03 – Editar equipa – A aplicação deve possuir a funcionalidade de edição de equipas. Isto implica, alterar as informações básica da equipa (nome, foto) e gestão de membros da equipa.
* RF04 – Listagem de equipas – A aplicação deve possuir a funcionalidade de listagem de equipas.
  + RF05 – Criar publicações – A aplicação deve permitir que os utilizadores consigam escrever publicações.
  + RF06 – Editar publicações – A aplicação deve permitir a edição do conteúdo das próprias publicações.
  + RF07 – Remover publicações – A aplicação deve permitir ao utilizador remover as próprias publicações.
  + RF08 – Agendar jogos – A aplicação deve permitir o agendamento de jogos de futebol.
  + RF09 – Editar jogos – A aplicação deve permitir a edição de jogos agendados durante um certo período.
  + RF10 – Listar jogos – A aplicação deve permitir a listagem dos jogos de futebol a que um certo utilizador está associado.
  + RF11 – Adicionar amigos – A aplicação deve possuir a funcionalidade de adicionar outros utilizadores como amigos.
  + RF12 – Remover amigos – A aplicação deve possuir a funcionalidade de remover a amizade de utilizadores adicionados previamente.
  + RF13 – Listar amigos – A aplicação deve permitir ao utilizador listar todos os seus amigos.
  + RF14 – Editar perfil – A aplicação deve permitir ao utilizador editar as suas informações básicas.
  + RF15 – Notificações – A aplicação deve possuir um conjunto de notificações que aparecem após o utilizador realizar uma certa ação.

### Requisitos não funcionais

Os requisitos não funcionais são especificações que descrevem aspetos da aplicação que não são diretamente relacionados às suas funcionalidades, mas que afetam o seu desempenho, segurança, usabilidade, compatibilidade e escalabilidade.

* RNF01 – Disponibilidade - A aplicação deve estar disponível 24 horas por dia, 7 dias por semana.
* RNF02 – Segurança - A aplicação deve ser segura e proteger os dados pessoais de todos os seus utilizadores.
* RNF03 – Desempenho - A aplicação deve ser rápida a carregar e responsiva ao toque do utilizador.
* RNF04 – Escalabilidade - A aplicação deve ser escalável para lidar com um grande número de utilizadores.
* RNF05 – Compatibilidade - A aplicação deve ser compatível com diferentes dispositivos e sistemas operativos.
* RNF06 – Usabilidade - A aplicação deve ser fácil de utilizar e navegar.

## Quality goals

A definição de metas de qualidades são uma das principais preocupações em qualquer projeto, especialmente em aplicações para dispositivos móveis e web. As metas de qualidade definem os padrões a serem seguidos e as expectativas a serem atendidas ao desenvolver uma aplicação. Estas metas ajudam a garantir que a aplicação seja fácil de usar, confiável, segura e escalável, entre outras coisas.

No caso de uma aplicação para agendar jogos amigáveis, as metas de qualidade são ainda mais importantes, pois é fundamental que a aplicação funcione de forma eficiente e sem falhas, para garantir a satisfação dos utilizadores. Além disso, é importante levar em consideração questões como segurança de dados, privacidade e compatibilidade com diferentes tipos de dispositivos, com a finalidade de proteger as informações pessoais dos utilizadores e garantir a melhor experiência possível.

Nesta seção, discutiremos as metas de qualidade que devem ser consideradas ao desenvolver uma aplicação para agendar jogos amigáveis.

|  |  |
| --- | --- |
| Quality Goal | Motivation/description |
| Usabilidade | A aplicação deve ser intuitiva e fácil de usar, com um design simples e elegante. |
| Confiabilidade | A aplicação deve funcionar de forma confiável, sem bugs ou falhas frequentes. |
| Desempenho | A aplicação deve ser rápida e responsiva, com tempo de carregamento rápido. |
| Segurança | A aplicação deve proteger as informações pessoais dos utilizadores de acordo com as melhores práticas de segurança. |
| Escalabilidade | A aplicação deve ser projetada para escalar com o crescimento do número de utilizadores, sem comprometer a performance. |
| Compatibilidade | A aplicação deve ser compatível com diferentes tipos de dispositivos, incluindo smartphones, tablets e computadores desktop. |
| Testabilidade | A aplicação deve ser projetada para ser facilmente testável, para que possa ser melhorada e corrigida rapidamente. |
| Facilidade de manutenção | A aplicação deve ser projetada de forma a ser fácil de manter e atualizar, sem necessidade de grandes esforços. |

Table 1 - Quality Goals

## Stakeholders

Os stakeholders são as pessoas ou grupos que têm interesses ou impactos no projeto ou empresa. Os stakeholders incluem desde os usuários finais até os investidores, passando por desenvolvedores, empresas de tecnologia, autoridades regulatórias e parceiros comerciais.

Identificar e compreender os stakeholders é fundamental para o sucesso de um projeto. Cada stakeholder tem necessidades e expectativas diferentes, e é importante garantir que essas necessidades sejam atendidas durante o desenvolvimento da aplicação. Além disso, é importante estabelecer uma comunicação clara e efetiva com cada stakeholder para garantir que todas as questões relevantes sejam abordadas e resolvidas de maneira adequada.

No caso de uma aplicação para agendar jogos amigáveis, os stakeholders incluem utilizadores finais, desenvolvedores, proprietários de campos de futebol, autoridades reguladoras e a comunidade em geral. Nesta seção, discutiremos os principais stakeholders de uma aplicação para agendar jogos amigáveis e exploraremos as suas necessidades e expectativas.

|  |  |
| --- | --- |
| Who? | Matters and concern |
| Developer | São responsáveis por criar e manter a aplicação, garantindo que atenda às necessi-dades e expectativas dos utilizadores. |
| Utilizadores | São os jogadores que usam a aplicação para agendar e jogar jogos amigáveis com seus amigos. |
| Parceiros comerciais | São empresas que trabalham em parceria com a aplicação para oferecer produtos ou serviços adicionais aos utilizadores. |
| Autoridades reguladoras | São responsáveis por regulamentar as aplicações, incluindo questões de segurança de dados, privacidade, etc. |
| Proprietários de campos de futebol | São os proprietários de campos de futebol que podem ser alugados pelos utilizadores da aplicação para jogar jogos amigáveis |
| Comunidade | São pessoas interessadas na aplicação, incluindo jogadores, entusiastas de desportos e defensores da vida saudável. |

Table 2 - Stackeholders

# Constraints

As restrições são fatores que limitam ou influenciam o desenvolvimento, estas incluem considerações técnicas, legais, de tempo e de recursos. É importante identificar e compreender as restrições desde o início do projeto para garantir que o projeto seja realizável e alcance seus objetivos.

As restrições técnicas são especialmente importantes pois podem afetar diretamente a funcionalidade e o desempenho da aplicação. Essas restrições incluem limitações no hardware, no software e na conectividade de rede, entre outras.

Nesta seção, discutiremos as restrições técnicas específicas da aplicação.

## Technical Constraints

|  |  |
| --- | --- |
| Constraint | Background and/or motivation |
| Hardware | A aplicação deve ser compatível com diferentes tipos de dispositivos, como smartphones, tablets e computadores. Isso significa que a aplicação deve ser otimizada para diferentes tamanhos de tela e configurações de hardware. |
| Software | A aplicação deve funcionar em diferentes sistemas operacionais, incluindo iOS, Android e Windows. Além disso, a aplicação deve ser compatível com diferentes versões dos sistemas operacionais para garantir a melhor experiência possível para os usuários. |
| Rede | A aplicação deve ser capaz de funcionar de forma eficiente em diferentes níveis de conectividade de rede, incluindo conexões Wi-Fi e dados móveis. |
| Segurança e privacidade | A aplicação deve proteger as informações pessoais dos utilizadores, incluindo nomes, endereços de e-mail, etc. Além disso, a aplicação deve ser projetada para ser segura contra-ataques cibernéticos e outras ameaças à segurança. |
| Integração com outros sistemas | A aplicação deve ser capaz de se integrar a outros sistemas, como calendários, mensagens de texto e notificações push, para fornecer uma experiência ao utilizador completa e integrada. |

Table 3 - Technical constraints

## Organizational Constraints

As restrições organizacionais são fatores externos que afetam o desenvolvimento de uma aplicação. No caso da aplicação a ser desenvolvida, algumas restrições organizacionais importantes a serem consideradas:

|  |  |
| --- | --- |
| Constraint | Background and/or motivation |
| Orçamento | O orçamento disponível para o projeto pode afetar a escala e o alcance da aplicação. Por exemplo, se o orçamento for limitado, pode ser necessário limitar o número de recursos incluídos na aplicação. |
| Prazo | O prazo para o projeto pode ser uma restrição importante, especialmente se houver uma data específica para o lançamento da aplicação. É importante garantir que o projeto esteja no cronograma e que as atividades sejam concluídas dentro do prazo estabelecido. |
| Recursos | A disponibilidade de recursos, incluindo equipe de desenvolvimento, pode afetar o desenvolvimento da aplicação. Por exemplo, se a equipe for pequena, pode ser necessário limitar o número de recursos incluídos na aplicação. |

Table 4 - Organizational constraints

## Conventions

|  |  |
| --- | --- |
| Constraint | Background and/or motivation |
| Design | Existem convenções de design para aplicativos móveis, incluindo padrões de navegação, cores e tipografia. É importante considerar essas convenções para garantir que a aplicação seja fácil de usar e atenda às expectativas do utilizador. |
| Tecnologia | Algumas tecnologias podem ser consideradas padrão em uma indústria ou organização. É importante considerar essas tecnologias ao desenvolver a aplicação, pois elas podem ser mais fáceis de implementar e manter. |
| Regulamentos e leis | Algumas regulamentações e leis podem estabelecer convenções para a coleta e armazenamento de dados. É importante garantir que a aplicação atenda a essas regulamentações e leis. |
| Padrões de segurança | Alguns padrões de segurança são aceitos como convenção em uma indústria ou organização. É importante considerar esses padrões ao desenvolver a aplicação para garantir a segurança dos dados do utilizador. |

Table 5 - Conventions

# Context

A aplicação a desenvolver permitirá a marcação de jogos de futebol com mais facilidade. Esta facilidade será garantida através das várias funcionalidades que a aplicação disponibilizará, das quais, um chat geral onde os utilizadores podem conhecer novas pessoas e porventura combinar jogos entre si. O objetivo é que este chat permita, então, que utilizadores descubram outros utilizadores dispostos a marcar jogos de futebol e por conseguinte enviar-lhes um pedido de amizade.

Com o referido anteriormente, os utilizadores podem enviar pedidos de amizade uns aos outros e, com isto, podem então adicioná-los a uma equipa ou agendar um jogo de futebol entre equipas. Cada equipa terão um capitão (administrador) que pode convidar novos utilizadores para fazer parte da equipa, remover utilizadores e agendar jogos de futebol entre equipas.

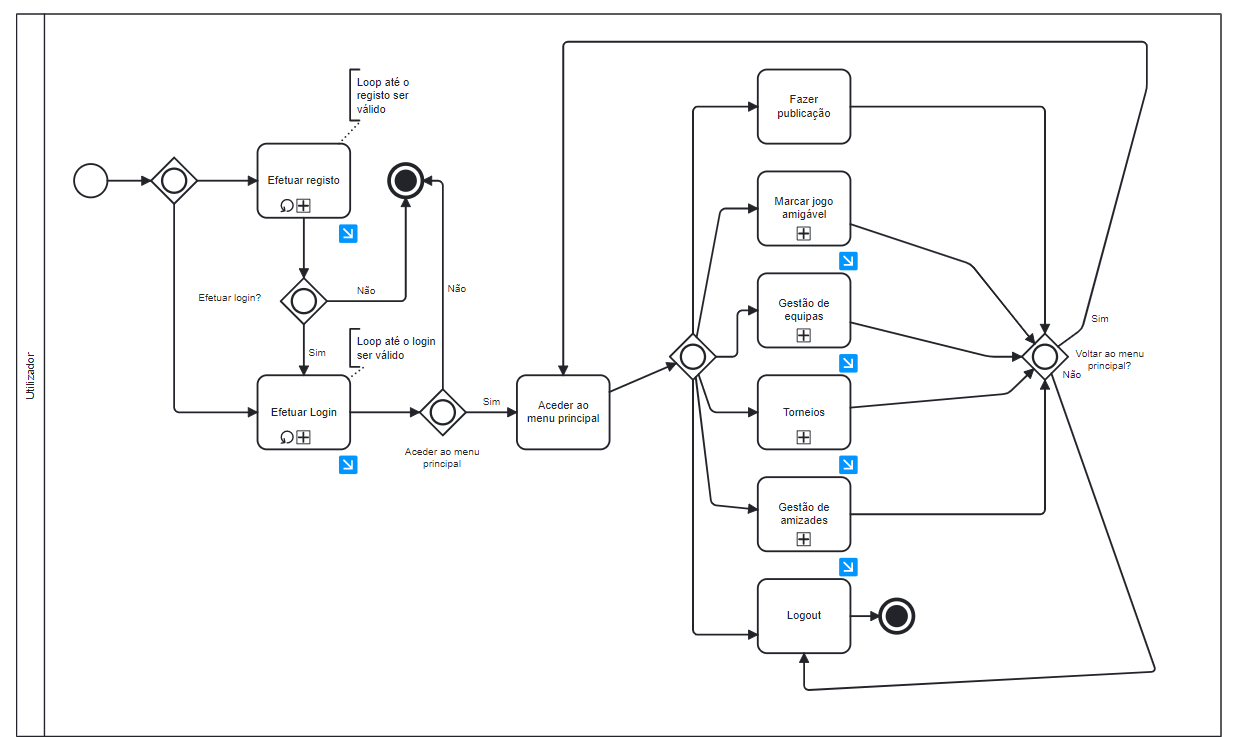
O agendamento de jogos é feito através de um menu onde o capitão propõe um jogo contra uma respetiva equipa e todos os jogadores de ambas as equipas serão notificados e devem aceitar ou recusar o jogo.

## Business Context

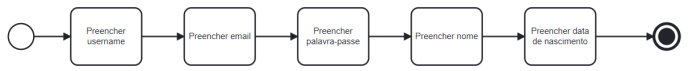
O contexto de negócios é uma consideração crucial ao desenvolver uma aplicação para agendar jogos amigáveis. Ele inclui aspetos como o mercado-alvo, a concorrência, a posição da empresa no mercado e o modelo de negócios. Compreender o contexto de negócios permite-nos desenvolver uma aplicação que atenda às necessidades do mercado e seja competitiva em relação a outras aplicações similares. Além disso, o conhecimento do contexto de negócio permite-nos concentrar nas áreas mais importantes da aplicação e evitar problemas desnecessários.

Para melhor compreensão do negócio foi elaborado um modelo bpmn como mostra na figura \*.

### Visão geral



### Registo



### Login

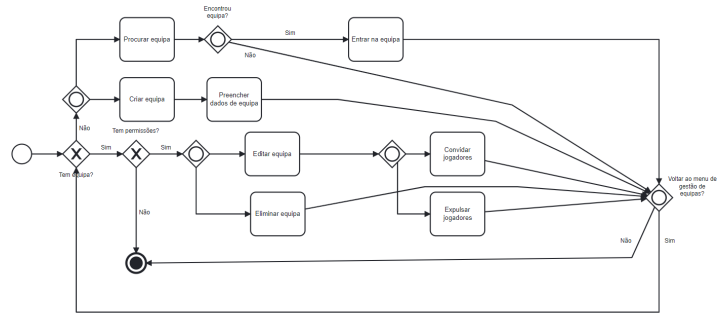
Uma imagem com texto

Descrição gerada automaticamente

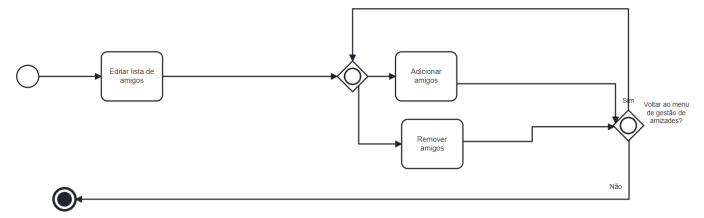
### Marcação de jogos amigáveis



### Gestão de Equipas



### Gestão de Amizades



## Deployment Context

Technical communication of the system with third parties

# Arquitetura aplicacional

Neste capítulo serão apresentados os capítulos referentes à implementação do projeto, tais como, o modelo arquitetural, a base de dados utilizada e o desenvolvimento dos micro serviços.

## Abordagem seguida

A abordagem seguida para o desenvolvimento do problema apresentado anteriormente passou pela implementação do modelo arquitetural apresentado na imagem seguinte.

image

Foram definidos três micro serviços onde cada um tem uma base de dados que armazena informações correspondentes à execução de funções respetivas ao próprio serviço.

Todos os micro serviços foram desenvolvidos utilizando a linguagem nodeJS tirando proveito da *framework* express que possibilita a fácil implementação de APIs REST.

Quanto ao nível de estruturação/arquitetura da aplicação foi utilizada a arquitetura *clean* não só porque também foi um requisito para a elaboração deste trabalho prático, mas também porque permite organizar o código de uma maneira onde o número de dependências é bastante reduzido.

### Base de dados

De modo a experimentar um novo tipo de base de dados, foi utilizada a MongoDB que é uma base de dados não relacional que segue uma estrutura de coleções onde cada coleção possui um conjunto de documentos. Tendo em conta o objetivo da cadeira decidimos implementar esta base de dados não localmente, mas sim na *cloud*. O facto de termos implementado a base de dados na *cloud* trouxe alguns constrangimentos quando tínhamos oportunidade de desenvolver o trabalho prático em aula, porque não era possível aceder à mesma através do código desenvolvido.

O serviço de alojamento da base de dados foi fornecido pelo próprio mongoDB que disponibiliza um cluster de forma gratuita.

### Serviço de autenticação

O micro serviço de autenticação é o responsável pelo acesso às restantes funcionalidades da aplicação, porque para o acesso a qualquer funcionalidade da aplicação é necessário efetuar login de modo que seja gerado um *token*.

### Registo

O registo na aplicação é efetuado através da rota “…/auth/register” e devem ser fornecidas um conjunto de informações no corpo do pedido.

Esta funcionalidade é uma das que tira proveito da biblioteca Bcryptjs, porque utiliza num registo é sempre necessário fornecer uma palavra-passe e, de modo a proteger esta palavra-passe aquando do seu envio para a base de dados, a mesma é encriptada utilizando os métodos genSalt e hash para a realização da encriptação.

Após a inserção do novo registo na base de dados é, também, inserido um novo log na base de dados com informações respetivas ao registo como, por exemplo, a hora da realização do mesmo, o nome do utilizador e o código respetivo à funcionalidade que foi executada.

Por fim, existem três valores de retorno possíveis após a execução da funcionalidade de registo:

* 200 – O utilizador foi registado na base de dados com sucesso.
* 401 – O registo falhou, porque o username escolhido para o registo já existe na base de dados.
* 404 - O registo falhou.

### Login

O login na aplicação é feito através da rota …/auth/login” e devem ser fornecidas um conjunto de informações no corpo do pedido.

A funcionalidade de login inicialmente recebe o nome do utilizador e a respetiva palavra-passe, depois faz um pedido à base de dados para verificar se o utilizador existe e para obter as informações relativas ao mesmo. Se este existir, é utilizada função *compare* da biblioteca Bcryptjs de modo a comparar a palavra-passe fornecida com a que está presente na base de dados. O modo de operação desta função é, basicamente, extrair o “salt” utilizado para encriptar a palavra-passe presente na base de dados e utiliza-o para encriptar a palavra-passe fornecida pelo utilizador.

Se o resultado da função *compare* for *true* é, então, gerado um novo *token* de acesso para o utilizador em questão e é gerado um *log* de login na base de dados.

Por fim, existem três possíveis valores de retorno desta funcionalidade:

* 200 – O utilizador iniciou sessão com sucesso.
* 403 – O utilizador forneceu uma palavra-passe inválida.
* 404 – O utilizador não foi encontrado na base de dados

### Alterar palavra-passe

Cada utilizador tem a possibilidade de alterar a palavra-passe através da rota “/auth/changepassword” onde deve ser fornecida a nova palavra-passe.

Para esta funcionalidade foi definido o pressuposto que o utilizador deve ter iniciada sessão na aplicação, porque, para a rota funcionar, o *token* deve estar presente no cabeçalho do pedido. Este requisito foi definido de modo a simular uma implementação em *frontend* e para não ter de enviar o *token* no corpo do pedido.

Inicialmente, é analisado o *token* recebido pelo utilizador e através dele é extraído o *username*. De seguida, é seguido um processo idêntico ao da funcionalidade de registo onde é, então, encriptada a palavra-passe utilizando os métodos função genSalt e *hash* da biblioteca Bcryptjs. Depois disso é feita uma *query* à base de dados que pesquisa pelo nome de utilizador obtido através do *token* e é substituída a palavra-passe antiga pela nova palavra-passe encriptada.

Por fim, esta funcionalidade apresenta dois códigos de erro possíveis:

* 200 – A palavra-passe foi alterada com sucesso.
* 404 – Erro.

### Microserviço Equipas

Um jogo de futebol não pode ser disputado sem existirem duas equipas e, como tal, foi necessário desenvolver uma rota responsável pela criação de equipas.

Na base de dados, na coleção das equipas, em cada documento é possível encontrar as seguintes informações de cada equipa:

* Id da equipa – gerado automaticamente
* Nome da equipa
* Data de criação
* O nome do capitão da equipa – Criador da equipa
* Membros da equipa

Esta funcionalidade pode ser acedida através da rota “…/teams/createteam” e recebe apenas o nome que o utilizador desejar dar à equipa, no corpo do pedido sendo que o *token* é obtido através do cabeçalho.

Aquando da realização do pedido, é obtida a data atual e o nome do utilizador é introduzido nos membros da equipa e no campo onde está explícito o nome do capitão da equipa.

# Interfaces Aplicacionais

Nesta secção iremos falar desde o logótipo da aplicação até ao design esperado da mesma.

## Logótipo

Como não tínhamos experiência nenhuma no desenho de logótipos, esta foi uma das fases mais complicadas de toda a unidade curricular, mas obtivemos novos conhecimentos porque decidimos trabalhar com o Adobe Illustrator.

O desenho do logótipo passou por várias fases, sendo que a primeira foi a de tentar transmitir a nossa ideia para o “papel”.

A nossa ideia de logótipo passou sempre por ter alguém a chutar uma bola e, inicialmente, queríamos que a bola fosse o “o” de *kickoff* tal como é possível ver pelas imagens abaixo.

Uma imagem com texto

Descrição gerada automaticamente

A próxima fase passou por tentar fazer uma versão mais realista do logótipo e, para tal, utilizamos uma foto de um jogador de futebol, escurecemo-la e tornamo-la num vetor.

Uma imagem com ClipArt

Descrição gerada automaticamente

No entanto, acabamos por desistir da ideia de tornar o logótipo mais realista, porque não estávamos a obter os resultados esperados.

Posto isto, decidimos dar um passo atrás e procurámos desenhar algo mais abstrato e, para tal, baseamo-nos num *template* do website Freepik e os primeiros resultado está presente na imagem abaixo.



Existiram alguns problemas na escolha de cores e, para contorná-lo, foram utilizadas combinações de cores de algumas equipas de futebol. No entanto, apenas foram escolhidas as duas últimas.

Uma imagem com transporte, ClipArt

Descrição gerada automaticamente

Inicialmente, após muita tentativa-erro, foi escolhida a versão cinzenta/branca com um toque ligeiro de azul, mas a conclusão obtida foi que esta combinação de cores podia causar um problema de cores na app devido à sua monotonia.

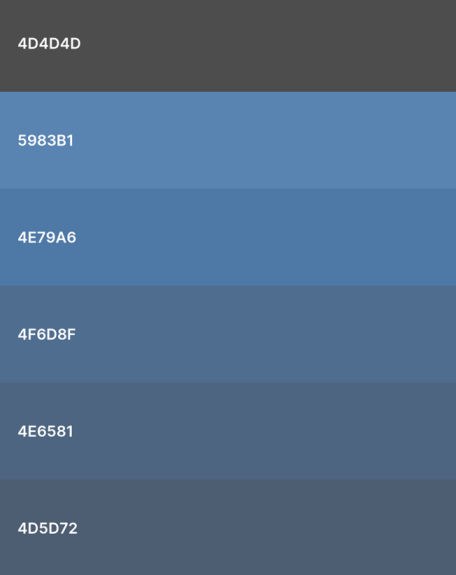


Por fim, decidimos avançar com a última combinação de cores que consiste numa mistura de azuis e cinzas.



### Paleta de Cores

As cores utilizadas para o desenho do logótipo e da aplicação foram as seguintes:



### Tipografia

A tipografia utilizada no desenvolvimento da aplicação foi a “Pilat Extended”. Esta fonte possui muitas variações e a maioria delas foi utilizada em diferentes áreas da aplicação.



Na imagem abaixo, é possível encontrar o tipo de fonte juntamente com as respetivas dimensões utilizado em certas áreas da aplicação.

Uma imagem com mesa

Descrição gerada automaticamente

## Análise de SWOT

Na imagem abaixo é apresentada uma visualização dos requisitos SWOT identificados para a Kickoff.

Uma imagem com mesa

Descrição gerada automaticamente

Abaixo é apresentada uma descrição de todos os pontos apresentados na imagem anterior.

### Pontos fortes

**Facilidade de utilização –** A aplicação foi concebida para ser de fácil utilização, permitindo aos jogadores agendar jogos e gerir as suas equipas de forma rápida e fácil.

**Unicidade do Serviço Oferecido** **–** Não existe no mercado uma aplicação com serviços semelhantes.

### Pontos fracos

**Disponibilidade geográfica limitada** **–** A aplicação só está disponível em certas regiões, o que limita o seu alcance e potencial base de utilizadores.

**Opções de personalização limitadas** **–** Alguns jogadores gostavam de ver mais opções de personalização, tais como a capacidade de criar calendários personalizados.

**Funcionalidade em desenvolvimento** **–**Como é uma aplicação relativamente recente algumas das funcionalidades não estão completamente desenvolvidas o que pode proporcionar más experiências de uso aos utilizadores e pode conter bugs.

**Falta de incentivo à competitividade** **–** Atualmente não existem torneios, nem recompensas que possam incentivar os utilizadores a competir uns com os outros.

### Oportunidades

**Expansão para novos mercados** **–** O aplicativo tem a oportunidade de se expandir para novos mercados e atingir um público mais vasto como, por exemplo, novos desportos.

**Parcerias com organizações de futebol –**: A aplicação poderia formar parcerias com clube/organizações de futebol para oferecer aos utilizadores mais recursos e benefícios como, por exemplo, vencedores de torneios podem ganham bilhetes para jogos de futebol.

### Ameaças

**Concorrência de outras aplicações** **–** Tendo em conta que no mercado não existe uma aplicação com um serviço semelhante, esta poderá vir a ser imitada.

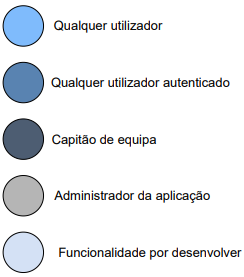
**Potenciais preocupações de segurança** **–** Como plataforma que lida com informações sensíveis, tais como dados dos utilizadores, a aplicação deve assegurar que as suas medidas de segurança são robustas e atualizadas para proteger os seus utilizadores.

## User Interface

### Sitemap

Um *sitemap* é um diagrama cuja finalidade é organizar e agrupar hierarquicamente as páginas de um certo *website* ou aplicação, de modo a facilitar a compreensão do mesmo. Além disso, também permite que o SEO (*Search Engine Optimization*) apresente a página num resultado mais favorável quando a mesma é pesquisada num certo navegador.

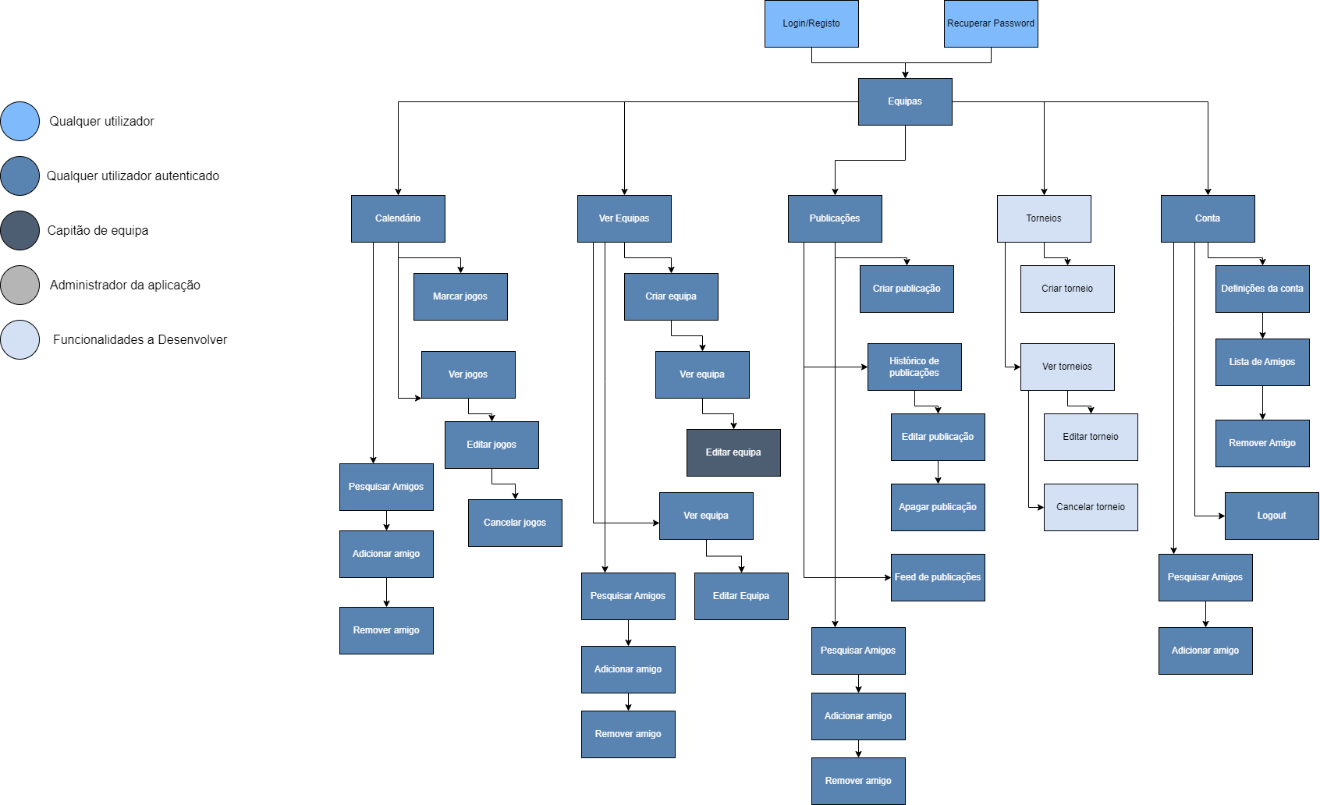
De modo a facilitar a compreensão do *sitemap* utilizámos uma combinação de cores onde cada uma corresponde a uma hierarquia de utilizadores. A legenda das cores pode ser encontrada na imagem abaixo:



Inicialmente, quando o utilizador inicia a aplicação, é lhe apresentada a página de *login* e, dentro da mesma, existe um botão que possibilita a recuperação da palavra-passe.

Quando o utilizador se autentica com sucesso, o mesmo é redirecionado para a *Homepage* e a partir dela consegue aceder às diferentes funcionalidades da aplicação, das quais, calendário, equipas, publicações, amigos e definições da conta.

Na imagem abaixo é possível encontrar o *sitemap* desenvolvido para a unidade curricular. Este diagrama foi desenvolvido utilizando o *website* [draw.io](file:///C:\Users\CARLOS\Desktop\draw.io).

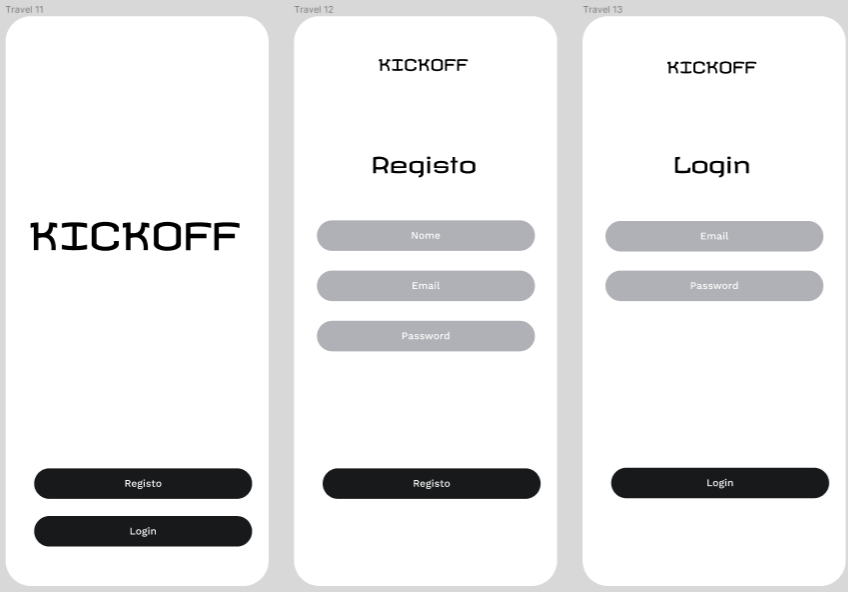


### Wireframes

Os *wireframes* são um protótipo da aplicação. São uma versão primitiva do visual final da aplicação cujo objetivo é apenas fornecer uma ideia aproximada de como será a interface, os elementos visuais e de como os utilizadores irão interagir com os elementos visuais.

Para o desenvolvimento dos *wireframes* da aplicação, utilizámos o [Figma](file:///C:\Users\CARLOS\Desktop\Figma.com) e tirámos proveito dos inúmeros *templates* disponibilizados. A maior parte dos componentes para a realização dos *wireframes* foram retirados do *template* [Wireframes Kit [Free]](https://www.figma.com/community/file/1122167340874425914).

Na imagem abaixo são apresentadas as janelas relativas à autenticação de um utilizador.



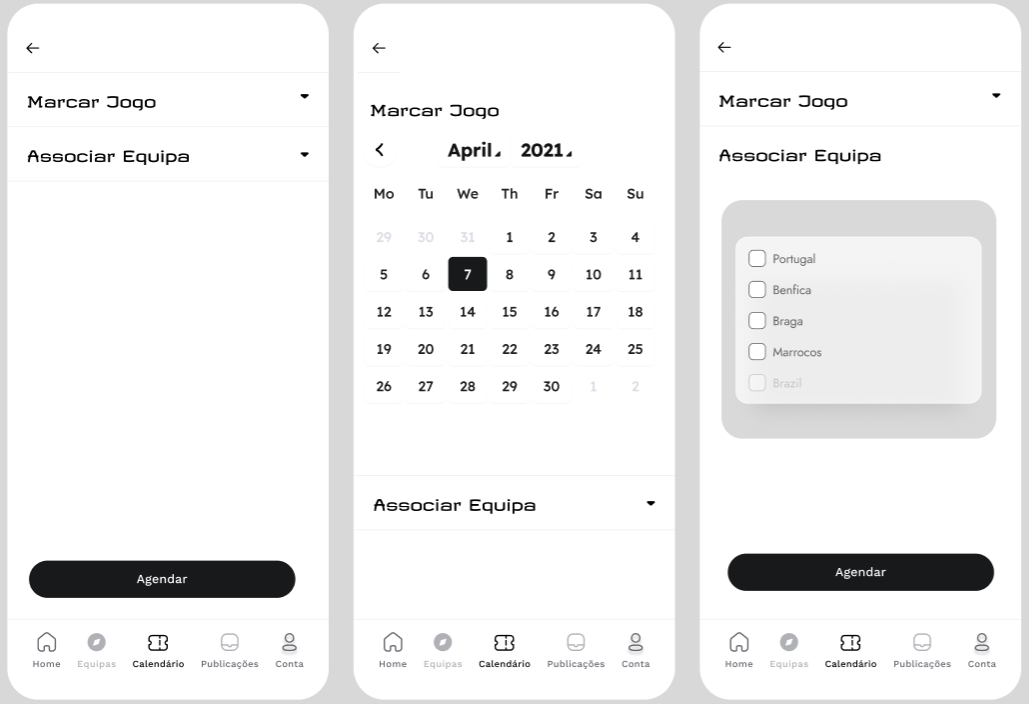
Como é possível verificar, existem as possibilidades de login e registo.

Nos *wireframes* relativos aos jogos desenvolvemos as funcionalidades de visualizar jogos já marcados, editar uma equipa e agendar um novo jogo.

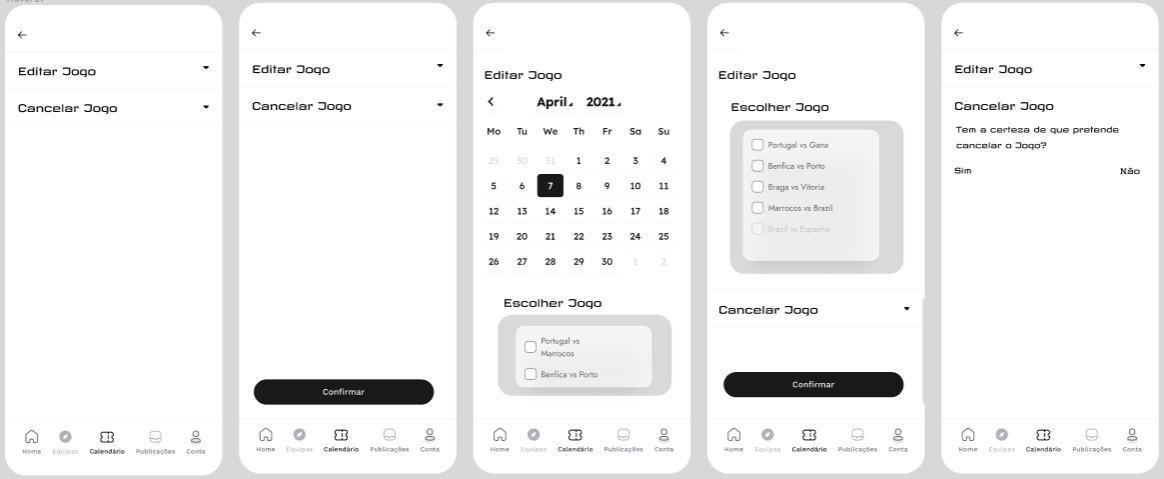
Para agendar um jogo o utilizador acede ao menu de calendário onde lhe é apresentado um botão que, quando pressionado, começa o agendamento de um jogo.



Quando o botão é pressionado, é apresentada uma nova janela com dois *dropdowns* respetivos às informações temporárias do jogo e a equipa que adversária.



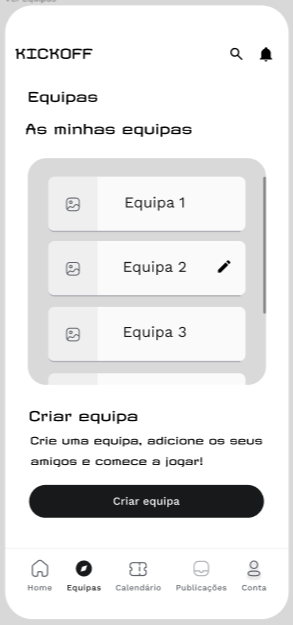
Como é possível verificar na imagem também existe um botão cujo objetivo é editar um jogo e, quando pressionado, existe a possibilidade de editar as informações relativas ao jogo ou cancelar o mesmo.



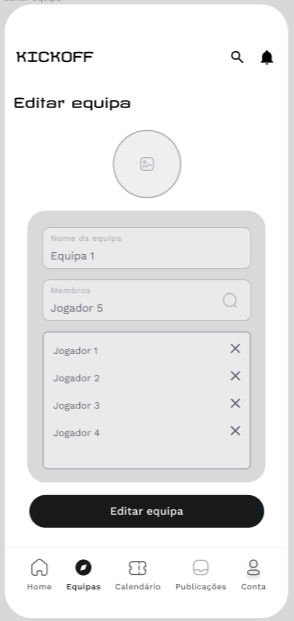
Relativamente aos *wireframes* das equipas, desenvolvemos uma janela responsável pela criação de uma equipa. O utilizador que cria uma equipa é definido como capitão da mesma.



Um utilizador, seja capitão ou não, possui um menu onde pode visualizar todas equipas de quais faz parte. Neste menu é, também, onde é fornecida a possibilidade da criação de uma equipa.



Como é possível ver pela imagem anterior, a “Equipa 2” possui um lápis à frente do respetivo nome. Isto indica que o utilizador atualmente autenticado é capitão da mesma e pode editar a mesma.



Além da opção de editar, que está apenas disponível para o utilizador capitão de cada equipa, todos os utilizadores presentes numa equipa podem aceder ao menu de equipa que é onde encontram as informações relativas à mesma, como futuros jogos e jogos passados.

Uma imagem com texto, comando, remoto, jogo

Descrição gerada automaticamente

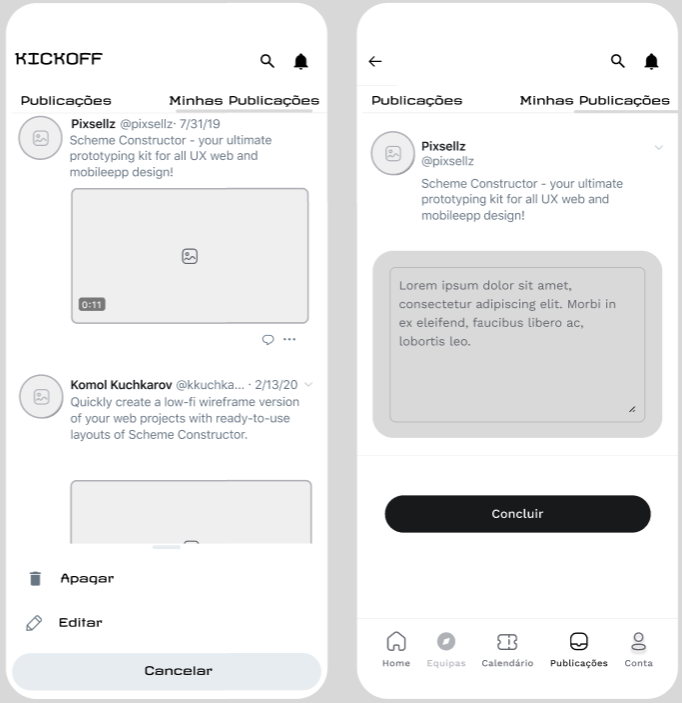
Existe também a possibilidade de os utilizadores escreverem publicações. Esta funcionalidade foi desenvolvida de modo a disponibilizar uma plataforma onde podem ser encontrados novos jogadores para jogar ou para criar uma equipa.

Como é possível ver pela imagem abaixo, existem duas abas. Uma que é relativa a todas as publicações realizadas por todos os utilizadores da aplicação e outra relativa apenas às do próprio utilizador de modo a fornecer um histórico de publicações.

Uma imagem com texto

Descrição gerada automaticamente

Existe também a possibilidade de editar uma publicação.



Ainda na aba de publicações, é possível encontrar uma lupa no canto superior direito. Esta lupa tem como objetivo fornecer a possibilidade de procurar por um utilizador em específico.

Uma imagem com texto

Descrição gerada automaticamente

Depois de encontrado um utilizador, podemos aceder ao perfil dele simplesmente clicando em cima do mesmo.

Uma imagem com texto

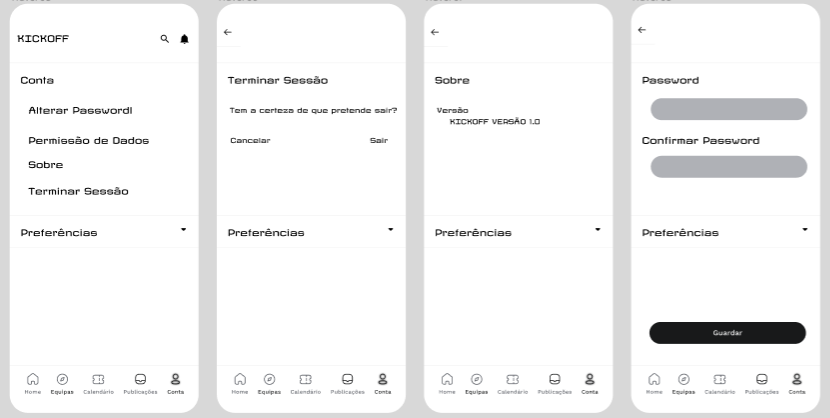
Descrição gerada automaticamente

De modo a aceder à própria lista de amigos, entrámos na área “conta” da aplicação. Como é possível ver podemos encontrar várias opções nesta aba.

Uma imagem com texto

Descrição gerada automaticamente

Relativamente a alterações de informações respetivas à própria conta, é possível realizá-las através das restantes opções: Conta e preferências.



Uma imagem com texto

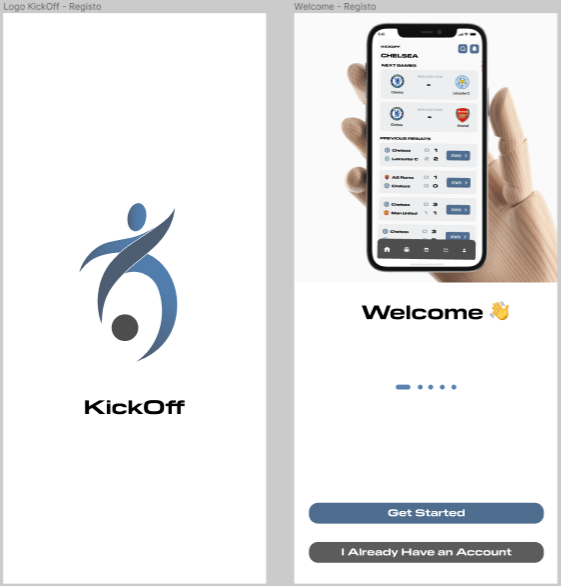
Descrição gerada automaticamente

### Mockups

Os *mockups* são, basicamente, um modelo ou uma replica da versão final de como é que o produto final será. Ao contrário dos *wireframes*, estes possuem um nível de detalhe mais elevado, de modo a alcançar uma melhor fidelidade com o produto final.

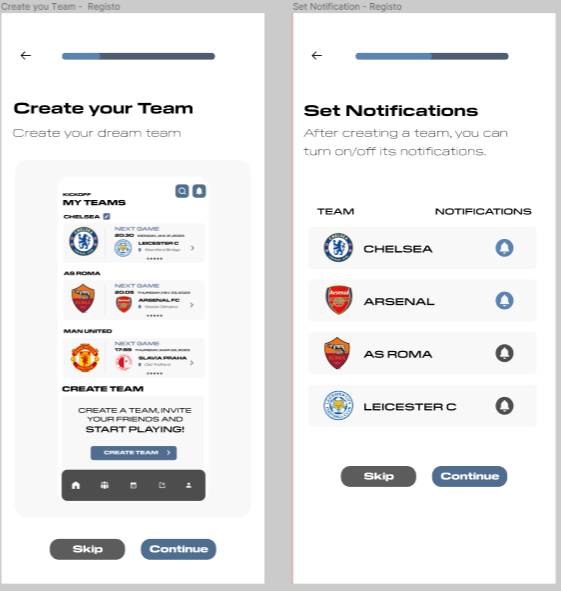
#### Autenticação

Inicialmente, quando o utilizador abre a aplicação, é apresentado o logótipo da Kickoff enquanto a aplicação está a carregar. De seguida, é apresentado o menu de boas-vindas onde o utilizador tem a possibilidade de efetuar *login* e efetuar registo.



#### Apresentação de funcionalidades

Decidimos adicionar algumas páginas de apresentação da aplicação, de modo a demonstrar aos utilizadores algumas das funcionalidades disponibilizadas. Estas páginas aparecem apenas quando o utilizador pressiona o botão de efetuar registo.



#### Registo

Após as páginas apresentadas anteriormente, são apresentados os menus de registo e, quando este é efetuado com sucesso, é apresentada uma página de sucesso e o utilizador é redirecionado para a aplicação.

Uma imagem com texto

Descrição gerada automaticamente

#### Login

Relativamente ao *login*, é um menu simples onde é pedido o *email* e a palavra-passe do utilizador. Além disso, existem as possibilidades de gravar a palavra-passe no *input* da mesma, de modo a descartar a necessidade de repetidas inserções e a possibilidade de recuperar a palavra-passe caso esta seja esquecida.

Uma imagem com texto

Descrição gerada automaticamente

#### Recuperar palavra-passe

Para recuperar a palavra-passe, o utilizador pressiona o botão correspondente a esta funcionalidade e, de seguida, é apresentado um menu onde o mesmo deve inserir o *email* respetivo à conta a qual pretende recuperar a palavra-passe.

De seguida, é enviado um código para o respetivo *email* e o utilizador deve inseri-lo. Caso este código seja validado, é apresentado outro menu para o utilizador definir uma nova palavra-passe.

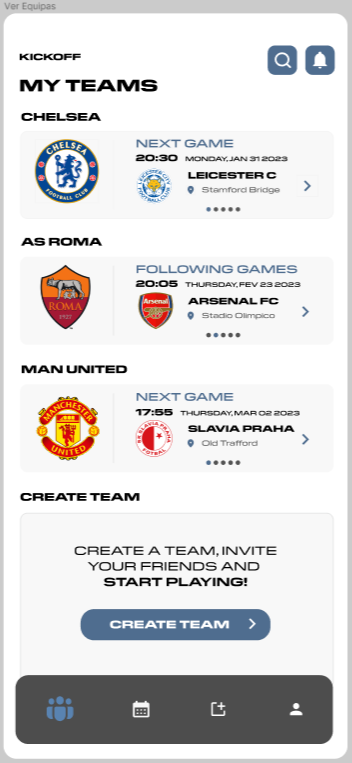
Uma imagem com texto

Descrição gerada automaticamente

#### Equipas

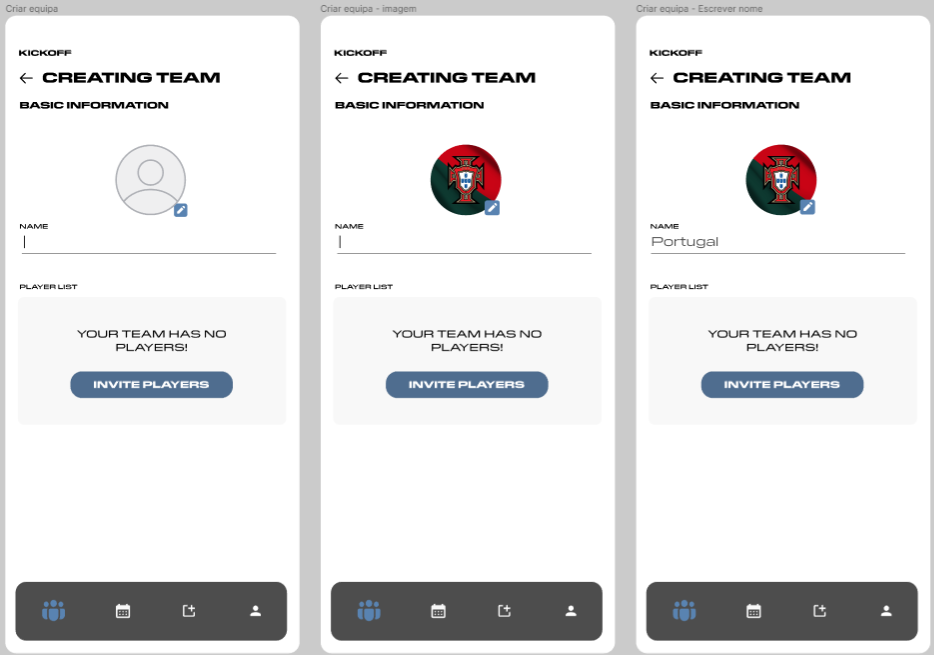
A primeira página que o utilizador vê quando inicia sessão com sucesso é a página das equipas. Nesta página é possível encontrar as equipas a quais o utilizador pertence e os próximos jogos das mesmas através do carrossel à direita do logótipo de cada equipa.

Além disso, também é apresentado um botão para a criação de uma equipa.

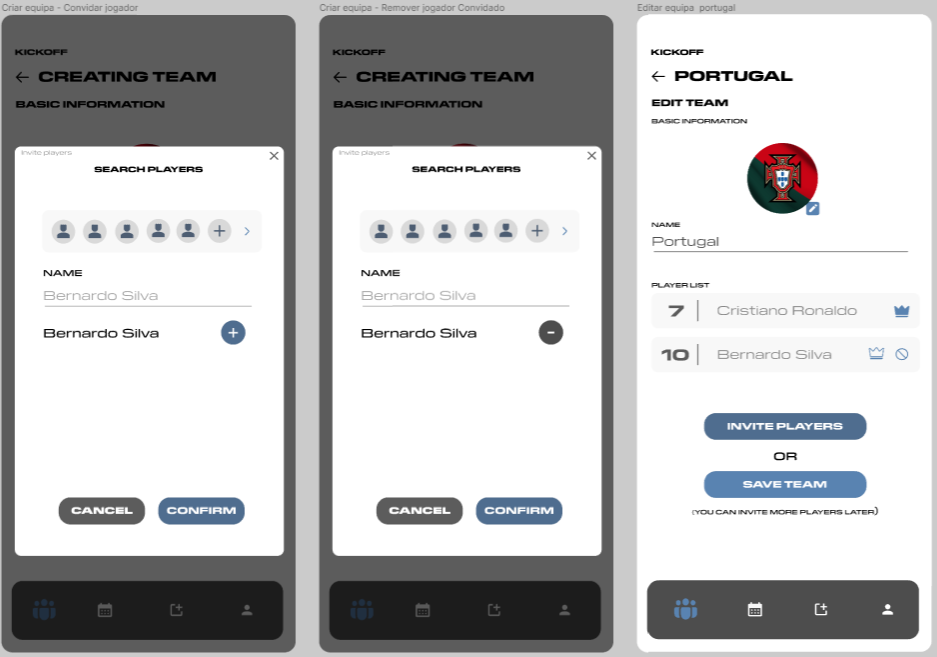


#### Criar Equipa

Após pressionar o botão de criação de equipa da página apresentada anteriormente (X) o utilizador é redirecionado para a seguinte página.

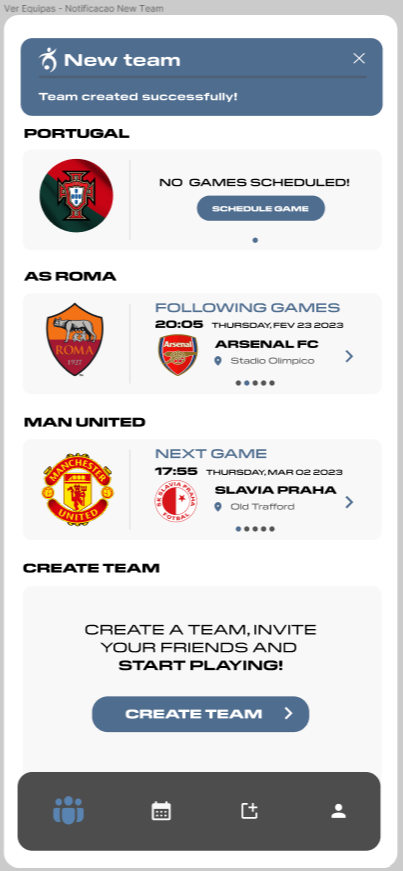


Após o utilizador preencher o nome da equipa e adicionar o logótipo da mesma, o mesmo pode enviar convites a outros utilizadores para se juntarem à equipa.



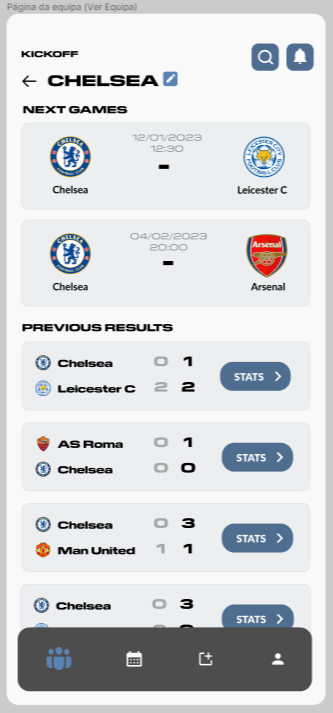
Como é possível ver pela página mais à direita da imagem anterior, existem dois jogadores presentes na equipa apesar de ter sido adicionado apenas um, porque o Cristiano Ronaldo é o utilizador que criou a equipa.

No fim da criação da equipa o utilizador é redirecionado para a página principal e é apresentada uma notificação com uma mensagem de sucesso.

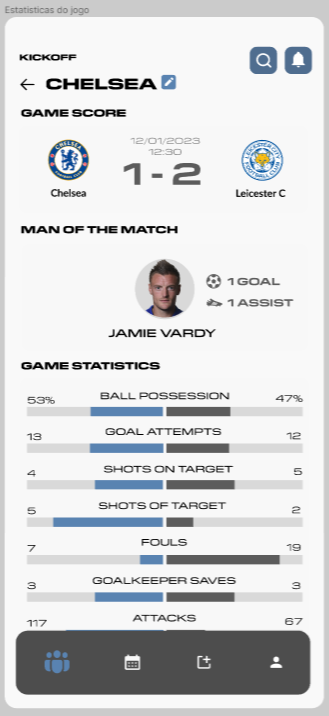


#### Página Equipa

Tal como nos *wireframes*, na página de equipa é possível ver os próximos jogos da respetiva equipa e alguns dos últimos resultados juntamente com as suas estatísticas.

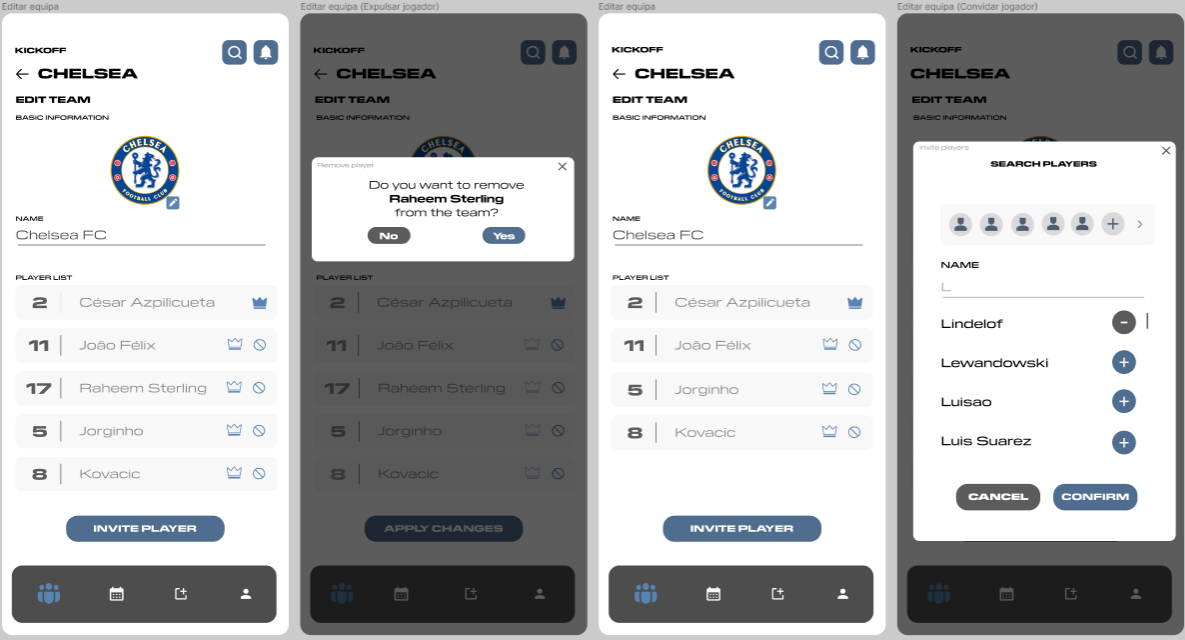


Quando um utilizador pressiona o botão “*Stats*” são apresentadas as estatísticas do respetivo jogo. Esta funcionalidade não estava desenvolvida nos *wireframes*.



#### Editar Equipa

Na Kickoff também é possível editar uma equipa após a sua criação. Como é possível verificar, é possível editar o logótipo, o nome e os jogadores.



Na imagem anterior é possível ver um jogador a ser removido da equipa e outro a ser convidado para a mesma.

#### Calendário

No calendário é possível selecionar um dia e visualizar que jogos estão próximos do dia selecionado.



#### Agendar um Jogo

Para agendar um jogo basta pressionar o botão “Schedule game” e o utilizador é redirecionado para a página apresentada na imagem abaixo.

Uma imagem com texto

Descrição gerada automaticamente

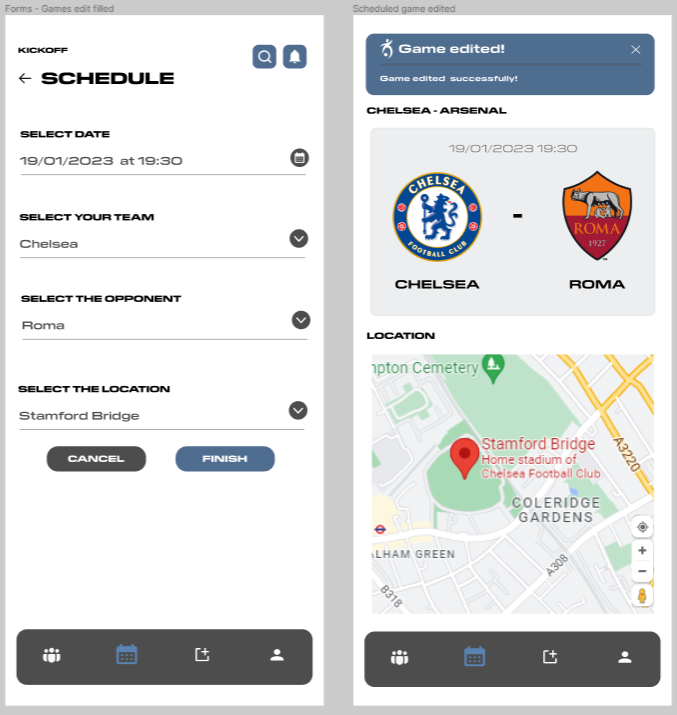
Após a informações estarem todas preenchidas, o jogo é agendado com sucesso.

Uma imagem com texto

Descrição gerada automaticamente

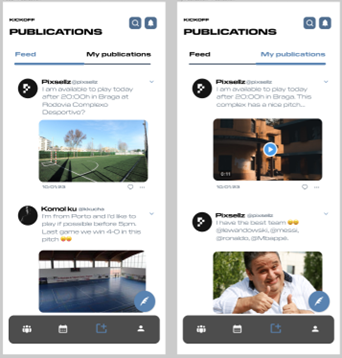
#### Editar jogo

Para editar um jogo basta selecionar o jogo na página do calendário e alterar as informações desejadas. Neste caso foi alterado o adversário.



#### Publicações

A área de publicações é composta por duas abas. Uma para o feed onde aparecem as publicações dos amigos do utilizador e outra para as próprias publicações do utilizador, esta aba funciona basicamente como um histórico de publicações.



#### Criar uma Publicação

Para criar uma publicação, basta pressionar o ícone da pena no canto inferior direito e aparecerá uma nova janela onde o utilizador poderá escrever a sua publicação. Nela poderá também anexar imagens, criar sondagens e adicionar gifs.

Uma imagem com texto

Descrição gerada automaticamente

#### Editar publicação

O utilizador poderá também editar ou remover publicações e para tal, basta simplesmente pressionar os três pontos no canto inferior da publicação e aparecerá um menu com as respetivas opções.

Uma imagem com texto

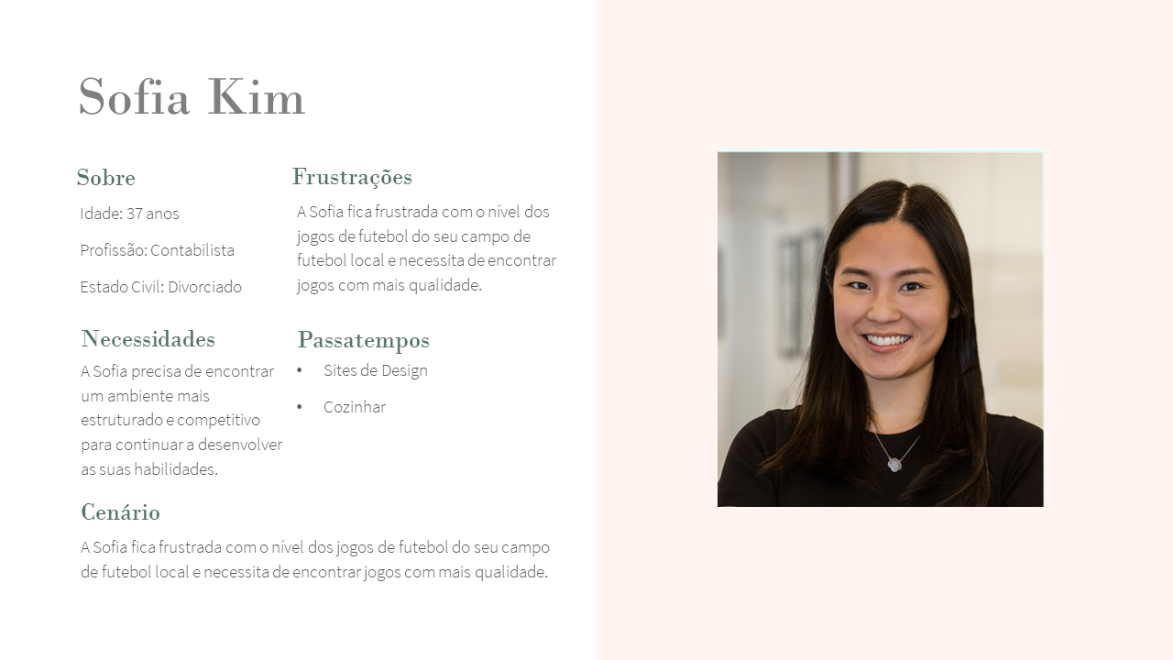
Descrição gerada automaticamente

## Personas e Cenários

As personas representam a forma de pensar e o comportamento de utilizadores. Estas personas não são pessoas reais, mas sim baseadas em utilizadores.

Relativamente aos cenários, servem para descrever a forma de como e para que é que os utilizadores utilizariam o futuro produto.

A Sofia Kim foi uma persona criada com o objetivo de representar um utilizador que pretende melhorar as suas habilidades futebolísticas.



O David foi uma persona criada com o objetivo de representar o utilizador que valoriza mais o aspeto social da aplicação.



O Jake foi uma persona criada com o objetivo de representar o utilizador que utiliza o futebol como uma forma de aliviar o stress.



A Rute foi uma persona criada com o objetivo de representar o utilizador que joga futebol profissionalmente e utiliza a aplicação para manter o ritmo competitivo.



## Teste de Usabilidade

Os testes de usabilidade são utilizados para obter informações sobre o desempenho de uma interfaceao nível da realização de certas tarefas.

Posto isto, foram desenvolvidos um conjunto de testes para obtermos conclusões sobre a aplicação desenvolvida. Estes testes foram desenvolvidos no *website*[Useberry](https://app.useberry.com/) e as tarefas definidas foram as seguintes:

* Criar de uma equipa
* Agendar um jogo
* Postar de uma publicação

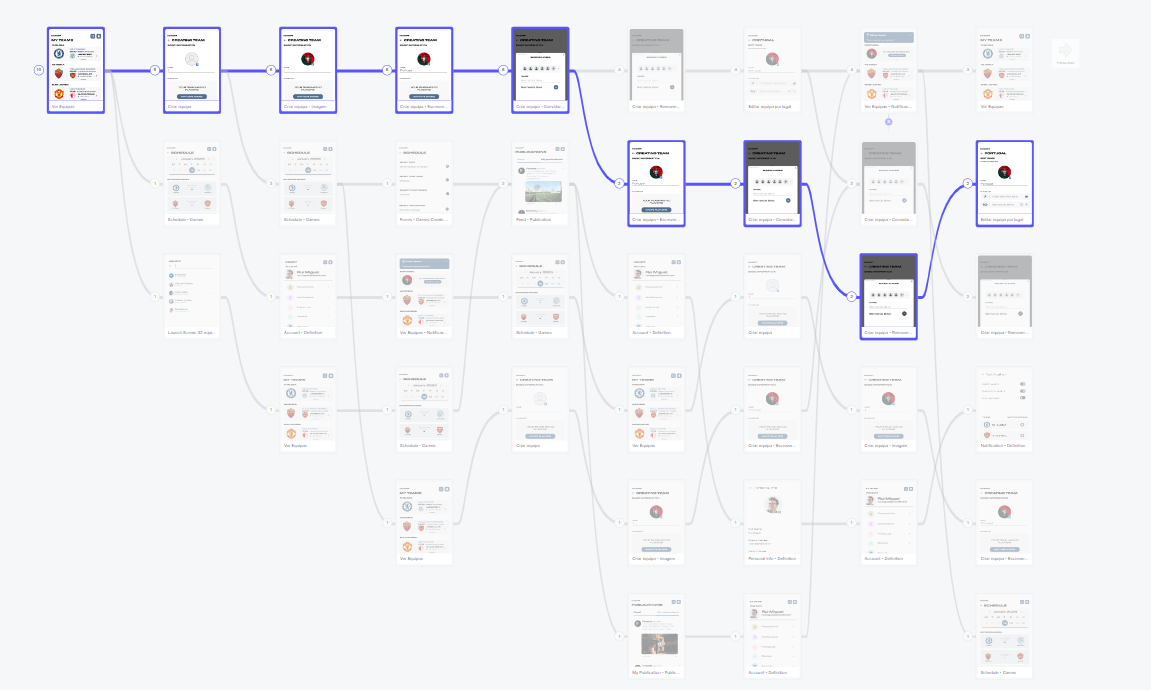
Os testes foram realizados por 10 pessoas e em 10 testes realizados, 9 foram positivos sendo que o único não positivo foi uma desistência antes da finalização de qualquer tarefa. A média de duração dos testes foi de 1m 41.4s.

Uma imagem com texto

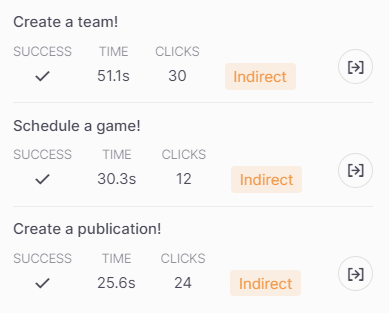
Descrição gerada automaticamente

A tarefa realizada com mais sucesso de acordo com as estatísticas foi a inicial, a de criação de equipa. Esta conclusão foi obtida através de estatísticas recolhidas no Useberry.

Na imagem abaixo, são apresentados todos os caminhos percorridos na realização de todas as tarefas e as janelas selecionadas representam o caminho percorrido mais frequentemente pelos utilizadores.



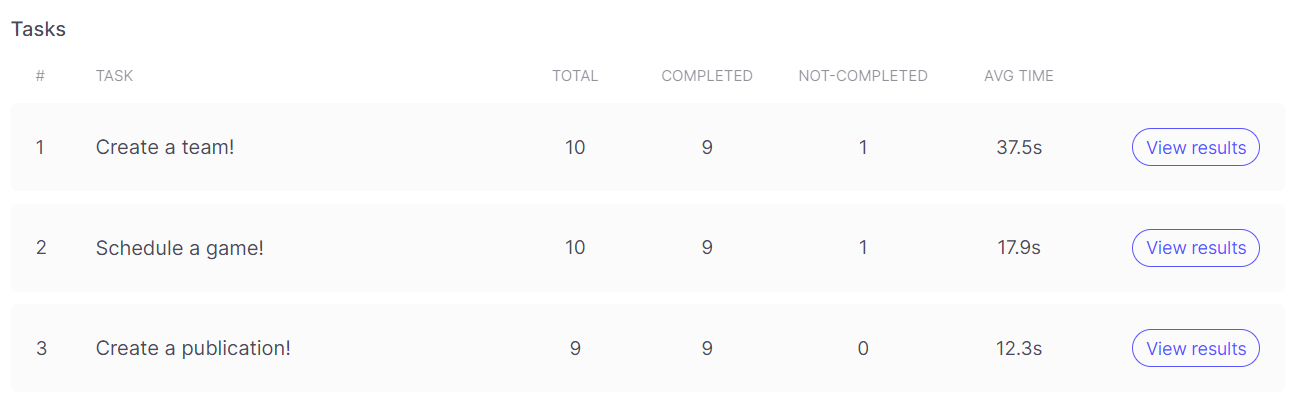
A conclusão obtida após analisar os resultados e os fluxos percorridos pelos utilizadores, foi que a aplicação possui vários caminhos para realizar uma tarefa. Isto pode ser observado ao analisar a execução do teste de um utilizador em específico.



Como é possível ver, nenhuma tarefa foi concluída percorrendo o fluxo direto, no entanto, todas as tarefas foram concluídas com sucesso.

Por fim, relativamente aos tempos médios de execução de cada tarefa, é possível observar que a tarefa de realização mais rápida por parte dos utilizadores foi a de criação de uma publicação. Isto pode ser explicado, pela familiarização com a aplicação obtida pela realização dos testes anteriores.

Figura 51 - Useberry media de tempos dos testes



# Design Decisions

(This section enables you to understand two fundamental design decisions of the system in detail. Optional Section)

## How Does the Engine Communicate with the Outside World?

### Problem Background

### Influences on the Decision

### Assumptions

### Considered Alternatives

### Decision

## Are Position Objects Changeable or Not?

### Problem Background

### Influences on the Decision

### Assumptions

### Considered Alternatives

### Decision

# Quality Requirements

(The quality scenarios in this section depict the fundamental quality goals (section 1.2) as well as other required quality properties. They allow the evaluation of decision alternatives.)

## Utility Tree

(Overview of the relevant quality attributes and their associated scenarios)

## Quality Scenarios

|  |  |
| --- | --- |
| ID | Scenario |
|  |  |
|  |  |
|  |  |

Table 9 - Quality Scenarios

# Glossary

## Terms

|  |  |
| --- | --- |
| Term | Definition |
|  |  |
|  |  |
|  |  |

Table 10 – Glossary terms

# Source Code

GitHub

# Sample Figures & Tables

CONTENTS OF THIS SECTION: Examples of figures and tables. Delete this page.



Figure 1: Sample Figure

Table 11: Sample Table

| Table Heading | Table Heading | Table Heading | Table Heading |
| --- | --- | --- | --- |
| Table Body | Table Body | Table Body | Table Body |
| Table Body | Table Body | Table Body | Table Body |
| Table Body | Table Body | Table Body | Table Body |
| Table Body | Table Body | Table Body | Table Body |

1. Appendices

|  |
| --- |
| CONTENTS OF THIS SECTION: Appendices may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data, API specification). As applicable, each appendix is referenced in the main body of the document where the data would normally have been provided. Appendices may be bound as separate documents for ease in handling. If your SAD has no appendices, delete this page. |

* 1. Heading 2 – Appendix title
  2. Heading 2 – Appendix tile