

Iago Figueira de Alencar

**UM APLICATIVO DE GERENCIAMENTO DE FILA  
RANQUEADA PARA ESPORTES CONVENCIONAIS GAMIFICADOS**

**PASSO FUNDO**

**2023**

Iago Figueira de Alencar

**UM APLICATIVO DE GERENCIAMENTO DE FILA  
RANQUEADA PARA ESPORTES CONVENCIONAIS GAMIFICADOS**

Projeto de pesquisa submetido ao Curso de Bacharelado em Ciência da Computação do Instituto Federal Sul-Rio-Grandense, Câmpus Passo Fundo, como requisito parcial para a aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II).

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
SUL-RIO-GRANDENSE - CÂMPUS PASSO FUNDO**

Orientador: Prof. Me. Élder Francisco F. Bernardi

**PASSO FUNDO**

**2023**

## ABSTRACT

*This article explores the development of a ranked queue management application for conventional sports, using matchmaking algorithms from electronic sports. Designed to enhance the sporting experience, the application employs performance analysis to ensure balanced matches, increasing competitiveness and participant engagement. The results demonstrate significant improvements in match fairness and user satisfaction.*

*Our research concludes that the application of gamification techniques in conventional sports is effective, highlighting the potential of integrating e-sports concepts for sports innovation. The study contributes to the field of sports gamification, providing a replicable model for similar initiatives, and paves the way for future investigations into the interaction between technology and sports.*

*Key-words: matchmaking; sports; ranked queue; games.*

## RESUMO

Este trabalho de conclusão de curso explora o desenvolvimento de um aplicativo de gerenciamento de fila ranqueada para esportes convencionais, utilizando algoritmos de matchmaking de jogos eletrônicos. O aplicativo, projetado para melhorar a experiência esportiva, utiliza análise de desempenho para garantir partidas equilibradas, aumentando a competitividade e o engajamento dos participantes. Os resultados mostram uma melhora significativa na equidade das partidas e na satisfação dos usuários.

A pesquisa conclui que a aplicação de técnicas de gamificação em esportes convencionais é eficaz, destacando o potencial da aplicação de conceitos de e-sports para inovação esportiva. O estudo contribui para o campo de gamificação esportiva, oferecendo um modelo replicável para outras iniciativas similares, e abre caminho para futuras investigações sobre a interação entre tecnologia e esportes.

*Palavras-chave: matchmaking; esportes; filas ranqueadas; jogos.*

## LISTA DE ILUSTRAÇÕES

Figura 1 — Esquema do funcionamento da arquitetura .....	14
Figura 2 — Diagrama de implementação.....	14
Figura 3 — Diagrama de sequência de acesso e busca de solicitação de recursos da arquitetura.....	20
Figura 4 — UI Flow .....	22
Figura 5 — Tela do matchmaking após a partida ser encontrada .....	24
Figura 6 — Tela do Manager .....	25
Figura 7 — Tela de perfil do usuário .....	26
Figura 8 — Animação do botão de início da procura.....	32
Figura 9 — Animação do botão de procura .....	33

## LISTA DE TABELAS

Tabela 1 — Definições das habilidades de um jogador .....	27
Tabela 2 — Definições da quantidade de jogadores nos grupos pré-definidos manualmente.....	30
Tabela 3 — Definições da quantidade de jogadores nos grupos pré-definidos pelo algoritmo.....	34
Tabela 4 — Definições da quantidade de jogadores nos grupos finais.....	34
Tabela 5 — Definições dos oponentes .....	35
Tabela 6 — Definição de elo em cada grupo final.....	36
Tabela 7 — Grupos que não formaram partidas.....	36

## SUMÁRIO

1	Introdução.....	6
2	Referencial Teórico .....	8
2.1	Matchmaking.....	8
2.2	Elo System.....	8
2.3	TrueSkill.....	9
2.4	E-sports.....	10
2.5	Filas ranqueadas .....	10
2.6	Trabalhos Relacionados.....	11
2.6.1	League of Legends.....	11
2.6.2	Appito .....	12
3	Metodologia.....	13
3.1	Aplicação proposta .....	13
3.1.1	Arquitetura da aplicação.....	14
3.1.2	Banco de Dados.....	21
3.2	UI Flow.....	22
4	Aplicativo .....	23
5	API.....	28
6	Avaliação.....	29
7	Resultados.....	32
8	Considerações finais.....	38
	Referências Bibliográficas.....	39

## 1 INTRODUÇÃO

Durante os últimos anos, os meios tecnológicos em cenários esportivos estão em ascensão. Se tratando dos esportes convencionais, uma área-chave de inovação que se apresenta em crescente notoriedade é a aplicação dos conceitos de matchmaking, originalmente aplicados aos esportes em ambientes virtuais ou “e-sports”, para o relacionamento no meio físico. Este desenvolvimento apresenta um marco importante na forma como os esportes tradicionais são organizados e praticados, enfrentando um desafio fundamental: como reunir jogadores eficazmente, considerando a disponibilidade e se baseando em critérios significativos?

A presença de jogadores na realização de esportes dentro do aspecto físico é uma característica intrínseca para que uma partida seja criada. No entanto, encontrar combinações de adversários à altura em termos de disponibilidade e habilidades, em diversos casos, torna-se uma tarefa frustrante. A experiência bem-sucedida do matchmaking para emparelhar jogadores com perfis semelhantes, através de algoritmos sofisticados, oferece perspectivas promissoras para solucionar este problema.

O objetivo deste artigo é a exploração da aplicabilidade de algoritmos matchmaking dos jogos eletrônicos nos esportes físicos, a fim de melhorar o gerenciamento e experiência dos jogadores, bem como, minimizar barreiras de acessibilidade à participação ativa nos esportes, desenvolvendo um sistema de matchmaking personalizado para esportes físicos.

A justificativa para este estudo reside na necessidade de inovação no campo do esporte. À medida que as expectativas dos participantes e espectadores continuam a evoluir, é importante encontrar formas de tornar as atividades desportivas mais inclusivas e convenientes aos seus usuários. A adaptação do matchmaking, uma técnica comprovada para jogos online, não só aumenta o envolvimento dos jogadores, mas também ajuda a descobrir novos talentos e a construir comunidades desportivas fortes.

Somada a isto, a pesquisa aborda um desafio interdisciplinar, combinando conceitos do esporte tradicional, ciência da computação e gestão esportiva. Ao relacionar estas questões, esperamos contribuir para o desenvolvimento de uma aplicação abrangente e eficaz ao matchmaking desportivo no ambiente físico.

Neste sentido, as hipóteses da pesquisa consideram que a aplicação de algoritmos de matchmaking em ambientes esportivos resultará em partidas mais equilibradas, justas e competitivas, em termos de habilidade dos jogadores, e que o sistema aumentará a satisfação dos jogadores e incentivará a participação contínua em esportes, assim como, que a associação dos jogadores dependerá da aplicação eficaz do matchmaking.

## 2 REFERENCIAL TEÓRICO

Esta seção está dedicada ao referencial teórico, onde ocorre a construção dos pilares e materiais bibliográficos para o desenvolvimento do presente trabalho.

### 2.1 Matchmaking

Um matchmaking representa o algoritmo de seleção de jogadores a fim de os relacionar para criação de uma partida. Diversos jogos como League of Legends, Gears of Wars, FIFA utilizam o matchmaking para o modo multiplayer de jogo. Entre os sistemas de classificação de jogadores para o matchmaking estão o sistema Elo e o *TrueSkill* [MINKA et al, 2018]. Esta Seção contempla conceitos importantes dos jogos que utilizam estes algoritmos.

### 2.2 Elo System

Introduzido pelo matemático Arpad Elo visando a definição da posição das habilidades dos jogadores de xadrez, a classificação do jogador no sistema Elo se dá pela estimativa da habilidade com o objetivo de definir a probabilidade de um jogador ganhar uma partida. O sistema introduzido por Elo leva em consideração a aplicação de uma distribuição Gaussiana para determinar a densidade da probabilidade, modelando as chances de ganhar ou perder uma partida [GRAEPEL e HERBRICH, 2006].

Desse modo, este sistema visa o ajuste da pontuação das habilidades para os jogadores, através do desempenho em determinada partida. A cada diferença de 400 pontos em relação a classificação, o jogador ou time com mais pontuações é apresentado com uma chance de vitória dez vezes maior em relação ao de menor pontuação, como mostra a expressão:

$$E_a = \frac{1}{1 + 10^{(R_b - R_a)/400}}$$

Conforme o usuário participa dos jogos, a quantidade de vitórias tende a 50% [VÉRON et al, 2014].

### 2.3 TrueSkill

O modelo para classificação de habilidades dos jogadores apresenta uma abordagem que assume uma pontuação para cada jogador, representando sua contribuição esperada dentro de uma equipe. O *TrueSkill* apresenta recursos para partidas com qualquer quantidade de equipes e qualquer quantidade de jogadores associados a estas equipes [HERBRICH et al, 2006].

Nesse método de classificação, a modelagem da pontuação dos jogadores para avaliar sua habilidade deve apresentar mutabilidade. Estas pontuações são atualizadas toda vez que uma partida é finalizada e toda vez que o sistema é iniciado porque o matchmaking assume que no final de cada partida, o usuário adquiriu uma experiência relevante a ser inserida no matchmaking. Desse modo, o algoritmo também treina o modelo e o classifica levando em consideração o tempo em que um jogador não participou de partidas [HERBRICH et al, 2006].

O *TrueSkill* acredita que o algoritmo de matchmaking deve garantir a proposta da partida, não permitindo com que jogadores mal-intencionados fujam do objetivo da partida para aumentar sua pontuação da habilidade no matchmaking. Independente do modo de jogo, a habilidade do jogador é considerada [MINKA et al, 2018].

Por questões de baixo custo computacional, a atualização das habilidades é executada em um servidor e armazenada em um banco de dados. Portanto, as representações para classificar uma habilidade no jogo não devem ser grandes e as atualizações não devem exigir muito do sistema [MINKA et al, 2018].

No modelo do *TrueSkill 2*, ou seja, uma versão atualizada, além das vitórias e derrotas do jogador, os dados estatísticos que dizem respeito ao modo como o jogador participou do jogo também é levado em consideração. Sendo assim, a forma com que um jogador contribuiu para equipe influencia no resultado da sua habilidade. Caso o jogador possua uma conduta positiva, ele é recompensado. Por outro lado, se sua conduta ou habilidades atrapalham de forma negativa, como abandono de partida, ele recebe uma penalidade que influencia negativamente na pontuação de habilidade. Ainda neste novo modelo, quando o jogador faz parte de uma equipe pré-definida, a habilidade dele é considerada maior do que o normal [MINKA et al, 2018].

## 2.4 E-sports

Os algoritmos de matchmaking são frequentemente utilizados nos *e-sports*. Os *e-sports* são uma forma de competição esportiva no meio digital através dos games eletrônicos. Neste cenário, os jogadores profissionais competem por uma premiação e usualmente são os melhores jogadores com base na classificação de habilidade do jogo em questão [MYŚLAK e DEJA, 2014].

Dentre as categorias do *e-sports*, destacam-se os *first-person-shooters* (FPS) como jogos de tiro em primeira pessoa, *fighting games* como jogos de luta, *real-time strategy game* como jogo de estratégia em tempo real, *Multiplayer Online Battle Arena* (MOBA) como game eletrônico no qual vários jogadores estão em uma arena e o objetivo principal é destruir a base inimiga [STEAM, 2022].

## 2.5 Filas ranqueadas

Para demonstrar as filas ranqueadas, é necessário apresentar as filas e os rankings no contexto dos games. As filas nos jogos são o resultado do tempo de espera em que um algoritmo de matchmaking system precisa para realizar as combinações necessárias entre os jogadores com o objetivo principal de criar uma partida. Já os rankings são classificações baseadas nas estatísticas dos jogadores. Dentro desta visão, pode-se afirmar que uma fila ranqueada é a execução do algoritmo de matchmaking para combinar os jogadores priorizando sua classificação no jogo [GAMES, 2022].

A classificação do usuário depende de uma série de fatores, embora todos levem em consideração a quantidade de vitórias, derrotas e, se possível, empates. Isto posto, uma vez que  $x$  é a quantidade de partidas ganhas e  $y$  a quantidade de partidas, quanto maior  $x$  for em relação ao máximo  $y$ , maior será o ranking. Portanto, fazendo o uso das filas ranqueadas, as partidas ranqueadas tendem a moldar uma pirâmide onde os melhores jogadores ficam no topo, neste sentido, o algoritmo entende que uma partida é moldada quando a combinação entre os jogadores é suficientemente justa. Os jogadores são recompensados quando ganham uma partida e punidos quando perdem ou performam mal ao participar de um jogo. Dessa forma, o jogador com a maior quantidade de pontos performou melhor em relação aos seus oponentes [MINKA et al, 2018].

## 2.6 Trabalhos Relacionados

Esta Seção está dedicada aos trabalhos que contemplam características semelhantes à aplicação proposta. Pelo fato de apresentarem características de matchmaking, definir jogadores com base na sua localidade e utilizarem de análise das habilidades dos jogadores para composição do time, as próximas seções contêm um exemplo de jogo online no estilo MOBA, denominado League of Legends, e o aplicativo Appito para gerenciamento de partidas de futebol no meio físico.

### 2.6.1 League of Legends

Ao abordar o game League of Legends, segundo [VÉRON; MARIN; MONNET, 2014] "Três fatores são cruciais para a experiência no jogo: tempo de espera, precisão da partida e o tempo de resposta do servidor."

Porém, primeiramente, é necessário explicitar algumas funcionalidades e conceitos do game. O League of Legends é um MOBA (Multiplayer Online Battle Arena), isto é, uma arena de batalha com dez participantes, onde cada jogador está vinculado ao time azul ou vermelho. Para participar do jogo, é definida uma rota (posição) e um campeão (personagem jogável). No primeiro momento, um jogador seleciona sua rota de preferência ao criar um grupo em um modo de jogo com a categoria normal game ou ranked game para a partida. Tal grupo varia de um jogador (O próprio player) até cinco jogadores (o próprio player e mais quatro integrantes) definido previamente para buscar uma partida. Uma vez que as rotas dos jogadores estão definidas e o líder do grupo inicia a fila de busca, ocorre o matchmaking da partida com o objetivo de relacionar os jogadores restantes para que uma partida seja criada. No segundo momento, com os dez jogadores definidos através do matchmaking, cada participante deve selecionar um personagem jogável [GAMES, 2022].

Um conceito de suma importância para a precisão da partida dentro do League of Legends se apresenta como elo. Criado pelo professor Árpád Élő, determinando um método estatístico de classificação das habilidades de jogadores de xadrez e posteriormente adaptado ao sistema LoL, fornece um cálculo com base nas habilidades dos jogadores e determina um ranking.

Para a formação de partidas, de acordo com os desenvolvedores do League of Legends, a base estabelecida se dá por: ranking do jogador, quantidade de partidas e a experiência de cada integrante do time [VÉRON; MARIN; MONNET, 2014]. Ainda neste sentido, é importante ressaltar que as equipes podem ser moldadas anteriormente e o matchmaking ser realizado através de equipes pré-estabelecidas, denominadas "premades". Por este motivo, em uma premade há a possibilidade de jogadores mais habilidosos serem inseridos em grupos que contenham usuários menos habilidosos. A experiência do usuário ajuda a corrigir este problema ao realizar uma matchmaking, no entanto, o valor de tempo de espera estimado da fila é elevado em 50%, aproximadamente. Embora ainda exista a possibilidade de uma partida estar desbalanceada [VÉRON; MARIN; MONNET, 2014].

Outro fator de suma importância está relacionado com a pontuação de MMR (Matchmaking Rating), em uma visão perfeita para o matchmaking, serão selecionados dez indivíduos com o mesmo MMR ao mesmo tempo, onde cada participante joga com a posição definida na premade, como visto anteriormente. No entanto, a realidade demonstra a necessidade de formar equipes que não atingem os requisitos ideais. Portanto, trata-se de filtrar o matchmaking levando em consideração uma partida justa, preferência de posição e filas rápidas [GAMES, 2022].

### **2.6.2 Appito**

O Appito é um aplicativo de organização para partidas de futebol, nele, é possível definir uma partida mensal automaticamente, convidar jogadores, controlar pagamentos por meio da carteira digital, encontrar jogadores em uma localidade com base no nível das suas habilidades, usar o Apple Watch para coletar dados sobre a performance [APPITO, 2022].

Neste aplicativo também é possível criar um perfil, registrar as estatísticas do jogo, receber notas e evoluir. É possível descobrir quais jogadores possuem maior nota dentro do grupo. O Appito trabalha com um sistema de recompensas e conquistas, isto é, quando um usuário atinge determinado objetivo, ele desbloqueia um rótulo em forma de conquista [APPITO, 2022].

### 3 METODOLOGIA

Para o primeiro passo do projeto, o objetivo é incluir os materiais bibliográficos para apresentar e fundamentar detalhes sobre o gerenciamento das filas ranqueadas, incluindo a arquitetura client-server, o protocolo HTTP, REST API's, Matchmaking e as tecnologias necessárias. Em um segundo passo, com base nos conceitos apresentados anteriormente, moldar o projeto definindo os usuários e, posteriormente apresentar a arquitetura da aplicação com as estruturas a serem inseridas, bem como a funcionalidade de cada uma na aplicação do método em jogos amadores. Por fim, a aplicação será testada para definir o desempenho dela em relação aos outros projetos existentes.

#### 3.1 Aplicação proposta

Esta seção contempla as funcionalidades e conceitos, determinando o modo como a aplicação lida com as necessidades. Portanto, para maior esclarecimento, é necessária a apresentação de alguns pontos. Dentro do sistema, a classificação de atores dentro dos casos de uso são: player, manager e *scout*. As classificações servem para atribuir as funcionalidades corretas dependendo do cliente.

A proposta da aplicação para jogadores está diretamente relacionada ao serviço de criação de partidas. Estes usuários participam do processo de matchmaking e podem ser inseridos no ranking. São atribuídas a estes personagens o *overall*, aplicando um valor com base nas suas habilidades. Existe a possibilidade de o jogador pertencer ou não a uma equipe.

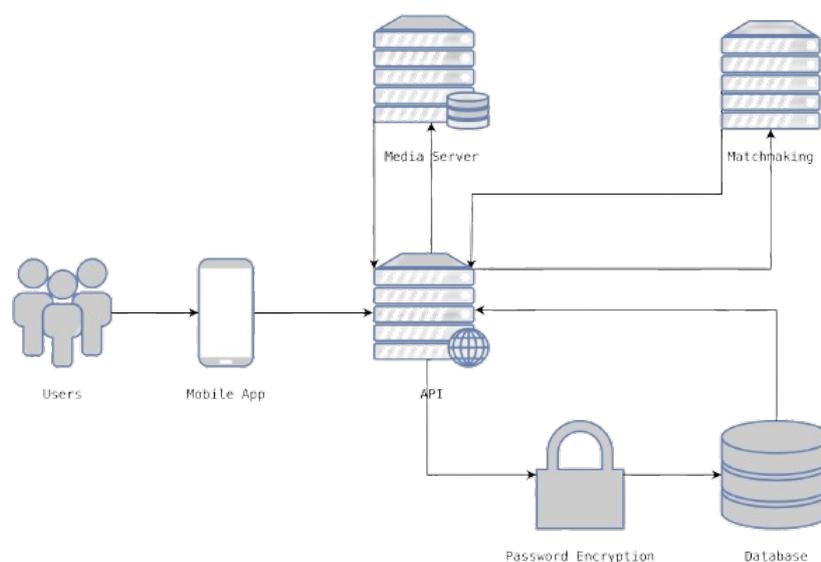
Dada a circunstância de que as tarefas visam o gerenciamento, as solicitações das filas ranqueadas podem acontecer individualmente, ou seja, um jogador pode solicitar o algoritmo sem estar vinculado a um grupo. Por outro lado, também é possível que este jogador participe de uma seleção de fila onde a busca leve em consideração uma equipe pré-definida e o matchmaking se aplique ao grupo projetado anteriormente. As funcionalidades, então, buscam atender o usuário com um ambiente que proporcione essa interação e métodos que incentivem este procedimento.

É de suma importância ressaltar que todas as funcionalidades buscam a gamificação do processo de busca por usuários ao gerenciar uma partida. Porém, a partida em si, acontece no ambiente físico.

### 3.1.1 Arquitetura da aplicação

Voltado para estabelecer a comunicação na aplicação, o objetivo é entregar uma arquitetura client-server dividida nos servidores de matchmaking e vídeo, bem como em um banco de dados e uma API fornecendo dados em JSON para autenticação do cliente, visualização das mídias e gerenciamento da fila no serviço de matchmaking. Então, através da aplicação, o foco é permitir ao usuário a participação individual ou em equipe de um algoritmo matchmaking que apresente sua habilidade como característica de definição para participação na equipe de forma equilibrada, a visualização estatística dos outros jogadores ou o gerenciamento de equipes. Os usuários podem interagir entre as partidas através de uma área social que os permite a visualização de vídeos, votação e troca de mensagens. As criptografias inseridas no banco de dados se trata somente de senhas para o registro e autenticação do usuário.

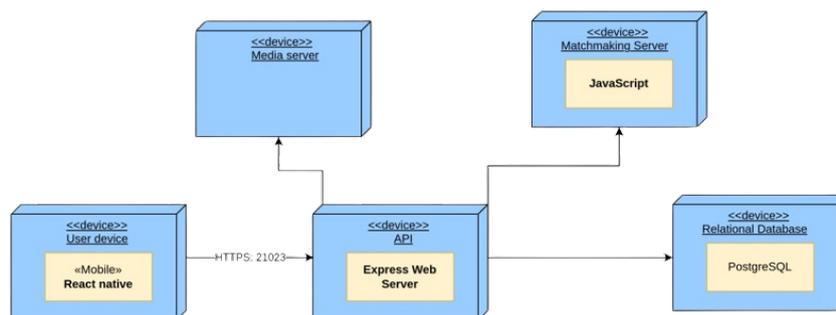
Figura 1 — Esquema do funcionamento da arquitetura



Fonte: Autor, 2023

A Figura 2 demonstra não só cada componente do sistema, mas também o fluxo de transferência de dados possíveis para quando uma requisição é feita pelo cliente. Desse modo, a API trabalha como forma de um meio para integração dos serviços e atender as necessidades do cliente.

Figura 2 — Diagrama de implementação



Fonte: Autor, 2023

Já a Figura 2 demonstra uma visão mais detalhada de cada componente do sistema. Portanto, temos que o dispositivo do usuário será uma aplicação móvel que realiza a comunicação através do protocolo HTTP, na porta "21023", a API será desenvolvida através do framework Express, o algoritmo de matchmaking será desenvolvido em PC em uma máquina e com banco de dados relacional utilizando PostgreSQL.

A seguir, o funcionamento da arquitetura tendo como pilar a Figura 2, Figura 3 e os requisitos funcionais para os casos de uso estabelecidos neste trabalho:

#### 1. Realizar Login:

- a) O usuário acessa a interface que contém as opções de envio das credenciais e as fornece;
- b) A API filtra se as credenciais são válidas e as envia para uma consulta no banco de dados;
- c) O banco de dados verifica os dados recebidos e retorna a consulta e as permissões para as credenciais para a API;
- d) A API, então, responde a solicitação do usuário com as informações retornadas pelo banco de dados e o concede acesso as demais funcionalidades da aplicação;

#### 2. Definir funções:

- a) O usuário logado na aplicação seleciona as funções que deseja ter dentro do sistema e as envia;

- b) A API recebe o pedido de atualização de funções e as encaminha ao banco de dados;
- c) O banco de dados armazena as novas funções do usuário;
- d) A API retorna ao usuário quais são as novas funções;

### 3. Buscar uma fila:

- a) O usuário logado na aplicação seleciona as funções que deseja ter dentro do sistema e as envia;
- b) A API recebe o pedido de atualização de funções e as encaminha ao banco de dados;
- c) O banco de dados armazena as novas funções do usuário;
- d) A API retorna ao usuário quais são as novas funções;

### 4. Conectar usuários:

- a) O usuário com permissão de *manager* convida um usuário com permissão de jogador para ingressar à equipe;
- b) A API faz a consulta no banco do jogador e verifica a disponibilidade de uma transferência ou por passe-livre e encaminha uma mensagem de solicitação de ingresso ao cliente;
- c) O cliente aceita a mensagem de convite e encaminha à API;
- d) A API vincula o usuário com permissão de jogador à equipe convidada através de uma atualização no banco de dados por CRUD e retorna ao cliente a nova equipe

### 5. Participar de equipe pré-definida:

- a) O usuário com permissão de jogador convida outros jogadores à participar do matchmaking;

- b) A API procura os usuários convidados no banco de dados e entrega uma solicitação caso estes estejam disponíveis;
- c) Os jogadores que receberam a solicitação devem aceitar o pedido de indexação à equipe pré-definida e o resultado é encaminhado à API;
- d) A API atualiza o registro da equipe no banco de dados e, a partir deste momento, o serviço de matchmaking consegue extrair os usuários em conjunto;

#### 6. Participar da comunidade:

- a) O usuário faz uma requisição das partidas relacionadas à categoria ao vivo ou melhores momentos;
- b) 2. A API busca os jogos que estão sendo transmitidos no servidor de video e gera um identificador no banco de dados caso este não tenha sido criado previamente ou solicita as informações de mensagens e quantidade de votos;
- c) O banco de dados retorna a informação do video;
- d) A API encaminha os dados ao cliente;
- e) A aplicação mostra ao usuário quais são os comentários e a quantidade de votos para todos os jogadores;
- f) A aplicação envia as mensagens do usuário à API;
- g) A API adiciona a mensagem à lista de mensagens do video no banco de dados;

#### 7. Visualizar ranking

- a) O cliente envia uma requisição definindo qual tipo de busca quer realizar
- b) A API solicita consulta os dados passados pelo cliente no banco de dados

- c) O banco de dados retorna os dados de determinado usuário
- d) A API encaminha ao cliente os resultados da consulta

#### 8. Visualizar partida:

- a) O usuário seleciona uma partida a ser visualizada através do seu identificador
- b) A API usa esse identificador para montar a consulta e o encaminha ao banco de dados
- c) O banco de dados retorna as informações disponíveis para determinada partida;
- d) A API encaminha os dados passados pelo banco de dados ao cliente;

#### 9: Transferir jogador:

- a) O usuário com permissão de manager faz uma solicitação de transferência à equipe que o jogador pertence
- b) A API consulta a equipe no banco de dados
- c) O banco de dados retorna ao usuário o identificador do dono para determinada equipe;
- d) A API encaminha os dados passados pelo banco de dados ao dono da equipe a que o jogador pertence;
- e) O usuário dono da equipe que contém o jogador recebe a solicitação de aceitação de transferência pelo valor determinado pelo usuário comprador e o aceita, então uma requisição é feita à API;
- f) A API faz o pedido de atualização ao banco de dados;
- g) O banco de dados atualiza as informações no registro e retorna à API;
- h) A API responde ao usuário que a transferência foi bem-sucedida;

#### 10. Votar em jogadores:

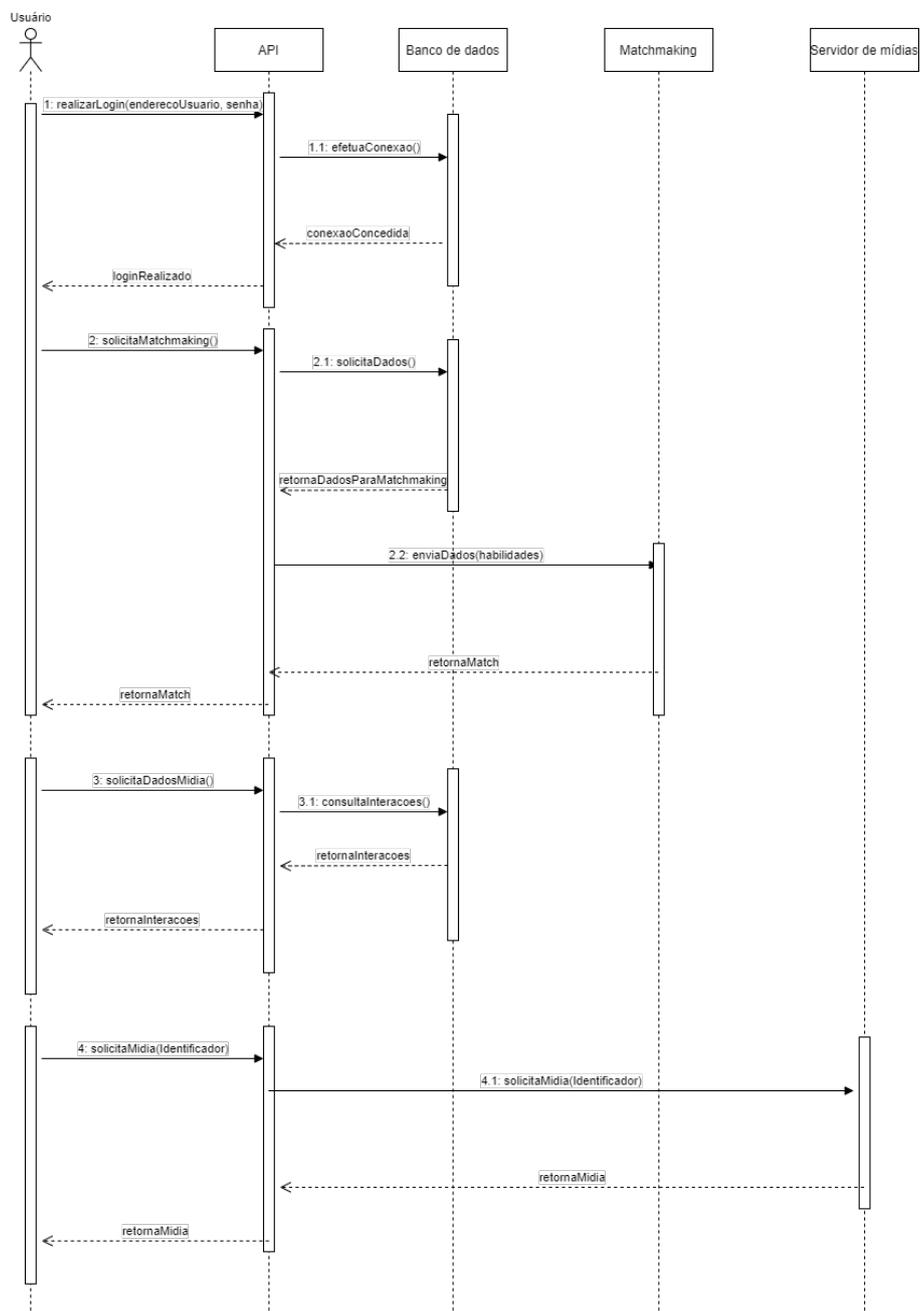
- a) Após o usuário receber a identificação da partida, é possível solicitar a votação, inserindo os dados a serem classificados para determinado jogador;
- b) A API usa esse identificador para montar a consulta e o encaminha ao banco de dados;
- c) O banco de dados atualiza as informações do usuário e registra em uma lista de usuários votantes;
- d) A API encaminha os dados passados pelo banco de dados ao cliente;

#### 11. Denunciar jogadores:

- a) Após o usuário receber a identificação da partida, é possível solicitar a penalidade ao jogador com conduta antiquada
- b) A API usa esse identificador para montar a consulta e o encaminha ao banco de dados
- c) O banco de dados atualiza as informações do usuário e registra em uma lista de denúncias;
- d) A API encaminha os dados passados pelo banco de dados ao cliente quando a denúncia for aceita e a penalidade aplicada;

Um diagrama de sequência 4 foi criado para melhor entendimento de como cada componente na arquitetura interage. Nela contém o fluxo de mensagens da comunicação dos objetos em relação aos casos de usos *Realizar Login*, *Buscar uma fila* e *Participar da comunidade*.

Figura 3 — Diagrama de sequência de acesso e busca de solicitação de recursos da arquitetura



Fonte: Autor, 2023

O front-end será desenvolvido em React Native e o back-end da aplicação será desenvolvido com o Node.js. O algoritmo de matchmaking está presente na API. Para o banco de dados relacional, a escolha é o PostgreSQL.

Agora, para os requisitos não funcionais, temos:

- **Mobile:** a aplicação será desenvolvida para dispositivos móveis através do React Native, devendo ser possível sua distribuição multiplataforma;
- **Armazenamento:** o armazenamento de dados do usuário deve ser criptografado em um banco de dados relacional;
- **Segurança dos Dados:** Garantir a segurança e a privacidade dos dados dos usuários, implementando criptografia de dados, autenticação robusta e medidas de proteção;
- **Usabilidade:** Interface intuitiva e amigável, garantindo que os usuários possam navegar e utilizar o aplicativo facilmente;

### 3.1.2 Banco de Dados

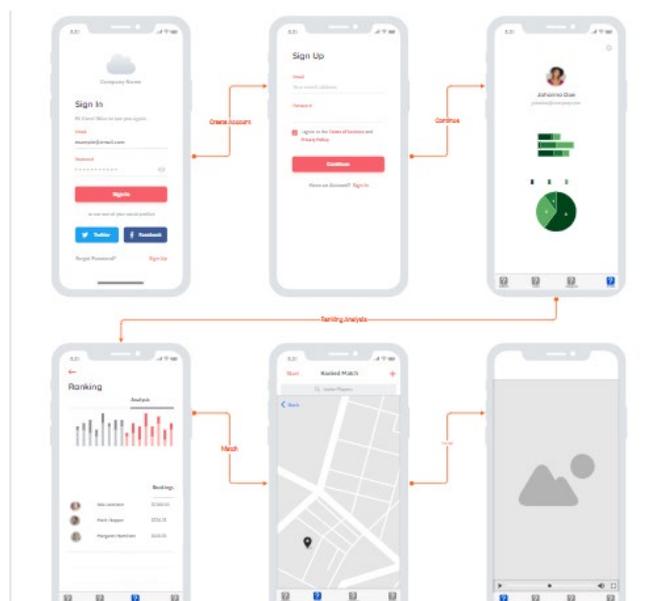
As entidades do banco de dados buscam o relacionamento entre as partidas, equipes e jogadores. Neste caso, um jogador se conecta à equipe pré-definida, contendo uma lista de outros jogadores. Em consequência, a equipe pré-definida está presente em uma ou mais equipes finais. Agora, para que uma partida seja realizada, a equipe pré-definida não pode participar de uma equipe final enquanto outra partida com uma equipe final que possua a equipe pré-definida ocorre. Neste sentido, a equipe pré-definida precisa finalizar a partida para que seja enquadrada em outro grupo final.

Visando a modelagem para a partida, esta relaciona duas equipes finais, por este motivo, há entidades na modelagem denominadas "opponents", "predefinedgroups", "finalgroups", "matches" e "players". Portanto, jogadores entram numa equipe base para formar times e, posteriormente, encontrar oponentes a fim de realizar o matchmaking, criando uma partida no ambiente físico.

### 3.2 UI Flow

A fim de definir um protótipo de como o usuário interage com a interface, foi elaborado um UI Flow. A Figura 1 contém o esboço para as telas de login, perfil, matchmaking visualização de ranking e mídia:

Figura 4 — UI Flow



Fonte: Autor, 2022

Seguindo o fluxo, a primeira tela sinaliza a tela em que o usuário realiza o login pelos campos de email e senha ou através de uma autenticação por outra aplicação. Já na segunda tela, o usuário consegue fazer a solicitação para registro no sistema. Uma vez que a autenticação do usuário é realizada, como demonstra a terceira tela, é possível acessar as funcionalidades. Ainda nesta tela, o usuário visualiza as suas estatísticas pessoais e as configurações da conta. Na quarta tela contém as estatísticas e ranking dos outros jogadores disponíveis na aplicação. A quinta tela representa um mapa com a opção de solicitar o matchmaking. Por fim, a sexta tela indica a interface de comunicação e mídia entre os usuários

## 4 APLICATIVO

O aplicativo foi desenvolvido através de *react-native*, considerando o Expo Go como ponto de simulação para que as *screens* sejam emuladas na tela do celular. Com o objetivo de visualizar o resultado do código, faz-se necessário a inicialização do projeto através de comando “*npm start*”, abrir o aplicativo do Expo Go e escanear o *Qr Code* disponibilizado na tela, considerando que ambos (*mobile e desktop*) estejam na mesma rede. Neste caso, todo o desenvolvimento foi elaborado com o auxílio de Android.

O aplicativo está dividido em três telas principais. A navegação do menu acontece por “*tab navigation*”. Da esquerda para a direita, o ícone de apito representa a *screen* de matchmaking. Nela ocorre a busca por usuários para que uma partida seja criada. Ao realizar interação com o botão, a função de disponibilidade do grupo pré-definido é disparada, significando que o grupo é elegível à participação do processo de matchmaking e pode entrar em partidas. A partir deste momento, a animação de busca aparece e, ao tocar, a função de busca é disparada novamente, porém, para afirmar que o usuário está cancelando a procura.

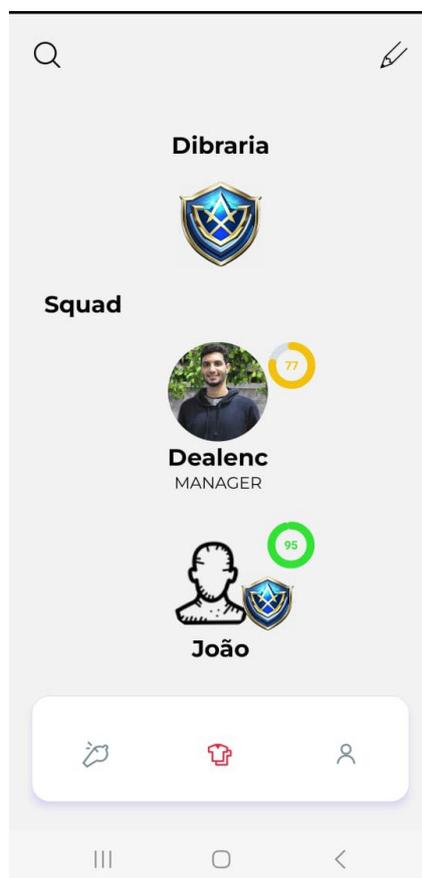
Figura 5 — Tela do matchmaking após a partida ser encontrada



Fonte: Autor, 2023

Quando a busca é efetuada, o app faz requisições a cada segundo à API envolvendo o matchmaking, dessa forma, quando o grupo pré-definido está inserido no grupo final e os oponentes já foram pareados, a *screen* atualiza e, em seguida, os participantes dos times vermelho e azul são mostrados de cada vez por carrossel. A fim de exibir os detalhes do grupo, é possível tocar na tela para que o modal esteja na tela e os times, vermelho e azul, sejam mostrados, em forma de lista. Esta estrutura, portanto, contém todos os jogadores e, desta forma, ao tocar no usuário, uma tela contendo as informações sobre as características do jogador é exibida.

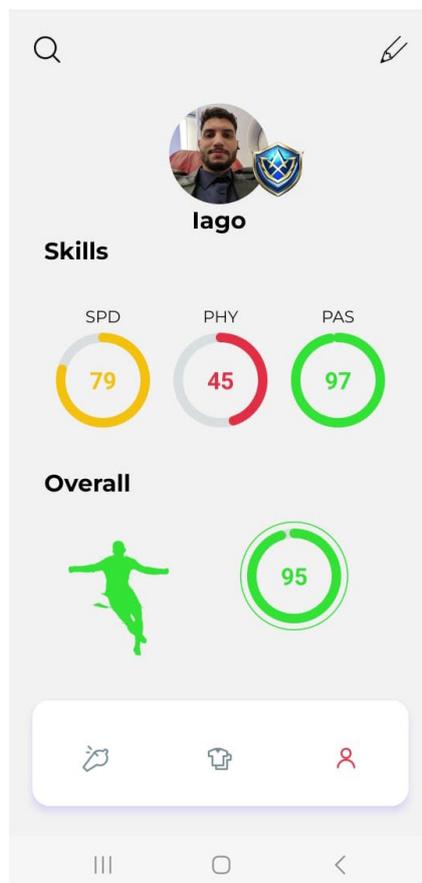
Figura 6 — Tela do Manager



Fonte: Autor, 2023

A tela do meio, por sua vez, é voltada para a área do técnico ou “manager”. Essa *screen* possui a finalidade de exibição das informações de um time. Desse modo, as estatísticas dos jogadores bem como a estatística do técnico e informações do time aparecem no momento da resposta da API em relação à requisição do usuário ao tocar no menu contendo o ícone com camisa. De cima para baixo, o aplicativo apresenta dois ícones, um para edição do time e o outro visando as configurações, embora não estejam habilitadas ainda, representam o header. Logo abaixo, existem os componentes do time, demonstrando seu nome por texto e respectivo elo por imagem do escudo, separados verticalmente. Em seguida, contém o *squad* com as características do manager, sendo elas: *overall*, nome e avatar. Posteriormente, além do técnico, a aplicação mostra um carrossel com todos os jogadores do *squad* disponíveis para o técnico.

Figura 7 — Tela de perfil do usuário



Fonte: Autor, 2023

Por último, a tela de perfil, com ícone de usuário, apresenta as características do jogador. Bem como o time, também possui um header para edição e configuração que está desabilitado. Em seguida, demonstra o avatar seguido por nome e posição do jogador no esporte. Ao clicar no avatar, ocorre a solicitação de permissão para que o aplicativo manipule os arquivos de imagens que estão no dispositivo móvel. Agora, existem duas categorias restantes: *skills* e *overall*. As *skills* são as habilidades do jogador, representadas da seguinte forma, como demonstra a Tabela 1.

Tabela 1 — Definições das habilidades de um jogador

<b>SKILL</b>	<b>SIGNIFICADO</b>	<b>TRADUÇÃO</b>
SHO	Shooting	Chute
SPD	Speed	Velocidade
PHY	Physics	Físico
PAS	Pass	Passe
DRI	Dribbling	Drible
DEF	Defense	Defesa

Fonte: Autor, 2023

Depois, há o *overall*, indicando a habilidade de referência média para o jogador na última categoria da *screen*. Cabe ressaltar que ambas as categorias são dispostas através de um carrossel com o *progress bar* em formato circular. Assim, para as habilidades que estão em um nível menor, ou seja, a habilidade do jogador é ruim, a cor de progresso é vermelha, já as habilidades que são regulares, o *progress bar* está em cor-de-laranja, e por último, as habilidades consideradas boas estão em verde.

## 5 API

A estrutura integra um banco de dados e uma API para a realização de matchmaking. A configuração do banco de dados é controlada por meio de um arquivo que determina se a aplicação está em produção e estabelece conexão com o banco de dados na nuvem ou local, dependendo do ambiente. As credenciais da AWS são armazenadas em um arquivo para suportar o armazenamento de imagens dos usuários no módulo S3.

No que diz respeito à modelagem do banco de dados, entidades são definidas com base em princípios de programação orientada a objetos. Funções são desenvolvidas para operações de consulta, inserção, atualização e exclusão de instâncias das entidades, além de funcionalidades específicas para o matchmaking e a visualização de estatísticas.

Os controladores gerenciam as funções da API, indicando sucesso ou erro de requisição por meio de códigos de status. As rotas da API são definidas para suportar os métodos HTTP GET, POST, PUT e DELETE. O matchmaking é implementado por meio de arquivos que configuram o algoritmo e garantem partidas equilibradas.

A API é configurada usando o framework Express, processando dados JSON e especificando a porta de execução. A função de matchmaking é inicializada para permitir a criação de partidas. Esse sistema completo é disponibilizado como uma API para ser utilizada em um aplicativo esportivo.

## 6 AVALIAÇÃO

O método de avaliação contempla artefatos que buscam examinar a qualidade do algoritmo de matchmaking e o comportamento do aplicativo ao lidar com interações do usuário e respostas da API. Neste sentido, para a classificação, foram utilizadas algumas abordagens.

Na tela principal de matchmaking do aplicativo, espera-se que nenhuma partida seja encontrada enquanto o usuário não pertencer a nenhum time que passou pelo algoritmo de matchmaking e está pareado. Nesta etapa, o grupo pré-definido a qual o jogador pertence está com a sinalização de que não procura partida, a animação do botão de busca intitulado “search” deve ser visualizada. Ainda se tratando do algoritmo de busca, enquanto o algoritmo não define os times, a expectativa é que a animação de busca tome a posição do botão “search”, esta alteração indica que algum jogador do grupo pré-definido solicitou a busca por partida. Por último, quando os oponentes forem selecionados incluindo o grupo pré-definido ao grupo final numa partida, espera-se a detecção e projeção dos times na tela. Como método de avaliação, um grupo pré-definido com a flag de disponibilidade desativada foi inserida. Neste sentido, o comportamento esperado é a ativação desta flag para que o grupo busque partida ao tocá-la.

Apesar de utilizar alguns conceitos de elo desenvolvidos por grandes jogos eletrônicos, o algoritmo não aplica conceitos importantes como a localização do usuário, por exemplo. No entanto, para a análise dos resultados e comportamento do matchmaking, os testes realizados se dão pela inserção de queries SQL diretamente no banco de dados, dessa forma, os usuários se relacionam apenas pelos grupos pré-definidos. Com a finalidade de examinar estas condições, foram gerados 67 inserções manuais de jogadores divididos em 15 grupos pré-definidos, incluindo o jogador visualizado no aplicativo. Estes estão dispostos conforme a Tabela 2.

Tabela 2 — Definições da quantidade de jogadores nos grupos pré-definidos manualmente

IDENTIFICADORES	QUANTIDADE DE JOGADORES
1, 2, 13, 14, 15	8
3	3
4	4
5	5
6, 7	2
8	7
9, 10, 11, 12	1

Fonte: Autor, 2023

Portanto, são oito jogadores nos grupos pré-definidos 1, 2, 13, 14 e 15, três jogadores no Grupo 3, quatro jogadores no Grupo 4, cinco jogadores do Grupo 5, dois jogadores nos grupos 6 e 7, sete jogadores no Grupo 8 e um jogador nos grupos 9, 10, 11 e 12. Estes deverão ser considerados aleatoriamente para que sejam gerados grupos finais.

Espera-se que grupos finais, através do processo de automatização e pareamento de jogadores, com a quantidade de jogadores máxima do esporte sejam tratados para a alocação em partidas, por meio da definição dos oponentes. Por este motivo, um grupo final não deve estar presente em duas partidas ao mesmo tempo e se encaixar em partidas com grupos finais que a variação dos elos seja maior que dois

Examinando a implementação do algoritmo na API, a lógica por trás do matchmaking está presente em um diretório específico. Consiste em dois arquivos: um arquivo destinado às configurações do matchmaking, como o intervalo em que a função que executa o algoritmo é realizada novamente, bem como a quantidade de jogadores disponíveis e tamanho do time para o esporte.

Sobre o arquivo que contém as funções que realizam o algoritmo de junção de times, primeiramente é chamada uma função que consiste em saber a quantidade de jogadores disponíveis, isto é, aqueles que habilitaram a flag de disponibilidade no app ou foram inseridos no banco de dados como disponíveis.

Se esta quantidade for suficiente para a criação de um time final, um grupo com os valores padrões são inicializados, ou seja, o overall, elo, tamanho do grupo até o momento, e ainda, flag para saber se a quantidade máxima de jogadores para o esporte foi atingida, flag voltada para a combinação de oponentes e se o grupo final está em partida. A quantidade de jogadores disponíveis na configuração é reiniciada e começa o processo de associação entre os grupos pré-definidos e os grupos finais. O algoritmo, então, busca todos os grupos finais que não têm o limite de jogadores atingido e todos os grupos pré-definidos disponíveis.

Para esta combinação, verifica-se a soma da quantidade de jogadores presente no grupo final e a quantidade de jogadores no grupo pré-definido são menores que o limite do esporte. Se esta condição se satisfazer, a função é chamada para determinar se a diferença do elo entre o time final e o grupo pré-definido é maior do que o limite permitido pelo aplicativo, atualmente padronizado como dois, conforme explicado anteriormente. Uma vez que o grupo pré-definido é o primeiro a ser inserido no grupo final, o elo do grupo final passa a ser o mesmo do primeiro grupo pré-definido. Entretanto, se já houver algum grupo pré-definido presente no grupo final, ocorre o ajuste do elo pela média entre o grupo final e o novo grupo pré-definido. Essa combinação entre grupo final e grupo pré-definido se dá pela entidade que relaciona o identificador de chave primária do grupo final e grupo pré-definido. Posteriormente, uma vez que o grupo final atingiu seu limite de participantes, a flag de limite de jogadores é ativada.

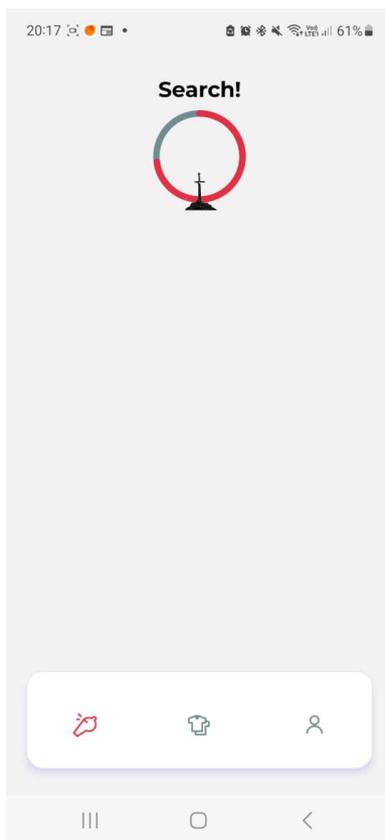
Agora, considerando que a quantidade de times com esta condição é maior que um e que o resultado seja um número natural, ou seja, dois times podem ser combinados com a finalidade de serem oponentes, são separados em "Red Team" e "Blue Team". Novamente, a função é chamada a fim de determinar se as habilidades entre dois grupos finais são balanceadas pelo padrão do algoritmo. Para satisfazer a condição de pareamento, faz-se necessária a apuração de Red Team e Blue Team com o objetivo de garantir que nenhum está em partida. Se todas estas etapas forem verdadeiras, os dois times estão aptos a serem oponentes e, então, a partida é criada.

## 7 RESULTADOS

Nesta seção contém os resultados obtidos pelas interações entre o usuário e a aplicação, bem como a visualização das respostas da API voltadas ao algoritmo de matchmaking e comportamentos do aplicativo. Primeiramente serão demonstrados os resultados da interação por busca da partida e, em seguida, a conduta do algoritmo com as inserções da Seção 6.

Com foco na flag de disponibilidade do grupo, o aplicativo iniciou com a flag desabilitada e não procurou por nenhuma partida durante o momento. Por este motivo, a animação do botão search esteve presente na tela enquanto não houve interações, como demonstra a Figura 5.

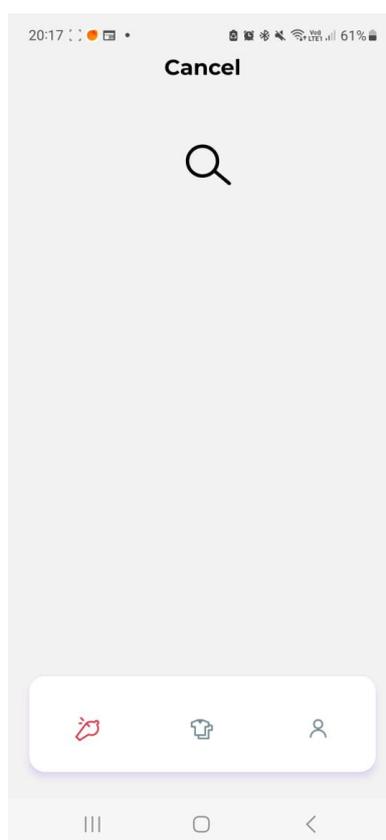
Figura 8 — Animação do botão de início da procura



Fonte: Autor, 2023

Considerando inicialmente que o usuário de teste que interagiu com a aplicação e estava presente em uma seleção de grupos pré-definidos somente para o seu time, o algoritmo relacionou corretamente a sua equipe a um grupo final e, posteriormente, à uma partida com um oponente válido e balanceado, segundo as especificações do algoritmo. A Figura 6 representa a animação da disponibilidade do grupo pré-definido.

Figura 9 — Animação do botão de procura



Fonte: Autor, 2023

Outro ponto de relevância é ressaltar que o consumo de dados entre a aplicação e a API permite a análise estatística através do perfil de cada usuário participante, como demonstra a Figura 4. Neste sentido, um usuário consegue visualizar quais participantes estão presentes na sua partida e, assim, comprovar que o seu ranking é compatível.

Se tratando da formação de oponentes, o papel do algoritmo de matchmaking é considerar todos os grupos pré-definidos disponíveis para participar do processo. Entre

eles, as quantidades de jogadores em cada grupo consideraram jogadores de quarto, sétimo e décimo elo nos grupos pré-definidos seguindo a Tabela 3.

Tabela 3 — Definições da quantidade de jogadores nos grupos pré-definidos pelo algoritmo

IDENTIFICADORES	QUANTIDADE DE JOGADORES
1, 2, 13, 14, 15	8
3	3
4	4
5	5
6, 7	2
8	7
9, 10, 11, 12	1

Fonte: Autor, 2023

Neste ponto, o algoritmo identificou corretamente a quantidade de jogadores relacionados à Tabela 2. Todos iniciaram com a flag de disponível ativa e flag de jogando desativada. Os resultados indicam que o algoritmo criou nove grupos finais, com a quantidade listada na Tabela 3.

Tabela 4 — Definições da quantidade de jogadores nos grupos finais

IDENTIFICADORES	QUANTIDADE DE JOGADORES
1, 2, 3, 4, 5, 6, 7 e 9	8
8	3

Fonte: Autor, 2023

Dentre eles, destaca-se o grupo com identificador 8, considerando três jogadores. Neste caso, o matchmaking não habilitou a flag de paridade, jogo e tamanho máximo atingido, portanto, significa que o grupo é elegível ao processo e a partir de novos grupos pré-definidos que encaixem à condição de formar um time completo, estes podem se adaptar aos grupos disponíveis para busca.

A Tabela 5 mostra como o algoritmo definiu os oponentes, considerando os grupos finais com oito jogadores.

Tabela 5 — Definições dos oponentes

IDENTIFICADORES	<b>RED TEAM</b> ( <i>Final groups</i> )	<b>BLUE TEAM</b> ( <i>Final groups</i> )
1	1	2
2	4	5
3	6	7

Fonte: Autor, 2023

Como primeiros oponentes, foram pareados o oponente em *red team*, o *final group* com identificador 1, enquanto para o *Blue Team*, o *final group* com identificador 2. Já para os segundos oponentes, o oponente em *red team* pertence ao grupo final com identificador 4, enquanto o identificador do *Final Group* 5 pertence ao *Blue Team*. Por último, o grupo final com a chave primária 6 é *red team* enquanto o grupo final com a chave primária 7 é *Blue Team*.

É de suma importância afirmar que após os oponentes serem formados, o algoritmo não considerou mais os grupos finais como elegíveis ao processo de matchmaking, consumindo a sua disponibilidade ao realizar a paridade dos oponentes e a criação da partida.

Por isso, ao investigar o resultado dos oponentes, os grupos que não foram relacionados afirmam o tratamento por parte do algoritmo de matchmaking quando eles diferentes se encontram. Para uma melhor observação, a Tabela 6 apresenta os identificadores de cada grupo final e o seu respectivo elo.

Tabela 6 — Definição de elo em cada grupo final

IDENTIFICADORES	ELO
1, 2	4
3	7
4, 5, 6, 7, 8 e 9	10

Fonte: Autor, 2023

Ainda, cabe ressaltar os grupos finais que não foram inseridos como oponentes e entender como o matchmaking os avaliou. Isso implica no modo em que o algoritmo define flags e trata times finais considerando o tamanho da equipe. Por este motivo, a Tabela 7 identifica o grupo final, o tamanho da equipe em números e flags, pareamento e a definição se o grupo está em partida.

Tabela 7 — Grupos que não formaram partidas

IDENTIFICADORES	FINAL SIZE	GROUP MATCHED	PAIRED	PLAYING
3	8	true	false	false
8	3	false	false	false
9	8	true	false	false

Fonte: Autor, 2023

Neste caso, o grupo final com identificador 3 não poderia formar partida com o grupo 8 porque a quantidade de jogadores do oitavo grupo final não permite, bem como o seu elo. Agora, entre o grupo 3 e 9, estes não poderiam formar partidas porque o resultado da diferença entre os elos também não é permitido. Por último, os grupos finais 8 e 9 não são oponentes porque a quantidade de jogadores no oitavo grupo não permite encaixar times com menos de oito jogadores.

Portanto, a validação do matchmaking se apresentou através da combinação de jogadores que se adequaram às necessidades e regras presentes na definição da formação de equipes. Os jogadores que buscavam por uma partida e atendiam os critérios foram inseridos em grupos finais e, posteriormente, a uma partida. No entanto, os jogadores que não foram combinados, precisam esperar para que novos jogadores com características semelhantes afirmem sua disponibilidade na aplicação.

Um problema enfrentado quando o algoritmo terminava a execução da função era constantemente procurar por novos grupos pré-definidos. Como solução para tal problemática está a presença da função de matchmaking sendo executada a cada segundo pela função assíncrona que estabelece a execução de matchmaking dentro de `setInterval` considerando um campo presente no arquivo, sendo 1.000 milissegundos.

## 8 CONSIDERAÇÕES FINAIS

Portanto, conclui-se que o balanceamento de equipes em competições esportivas, sejam elas virtuais ou físicas, é essencial para proporcionar experiências justas e competitivas aos praticantes. Os algoritmos de matchmaking utilizados em jogos eletrônicos, ou "e-sports", têm como objetivo formar equipes equilibradas com base nas habilidades dos jogadores, sua localidade e divisão, visando evitar disparidades e promover a igualdade de oportunidades. Além disso, a análise estatística dos jogadores e a promoção da comunicação entre os usuários são objetivos importantes do uso de algoritmos de matchmaking em esportes, contribuindo para uma experiência mais estratégica, envolvente e colaborativa. A arquitetura client-server é uma abordagem comum para a implementação de sistemas de matchmaking em jogos eletrônicos, permitindo a troca de informações entre o cliente (jogador) e o servidor (plataforma do jogo) para formar equipes equilibradas e proporcionar uma experiência satisfatória aos praticantes de esportes.

Com intuito de preservar o aplicativo, como possíveis atividades a serem implementadas em trabalhos futuros, destacam-se as autenticações de usuários, finalização e cálculo da pontuação na partida, acréscimo da busca por localização dos usuários, adição de chat e publicação de vídeos que promovem a interação da comunidade, edição de perfil do usuário, denúncia de usuários maliciosos, transferência de jogadores e segurança da aplicação. Desse modo, por se tratar do processo de relacionar jogadores, as funcionalidades implementadas seguindo o caso de uso são: buscar uma fila, participar de equipe pré-definida, visualizar partida e visualizar ranking.<sup>1</sup>

---

<sup>1</sup> Vale evidenciar que o presente trabalho, em formato de artigo, com o título “*Um aplicativo de gerenciamento de fila ranqueada para esportes convencionais*”, foi publicado no Encontro Anual de Tecnologia da Informação - EATI, promovido conjuntamente pela UFSM (Universidade Federal de Santa Maria) e IFFar (Instituto Federal Farroupilha), podendo ser acessado através do link: <<https://eati.info/trabalhos/apresentacoes/>>

## REFERÊNCIAS BIBLIOGRÁFICAS

APPITO. Appito App. 2022. Disponível em: <<https://appito.com/app/>>. Acesso em: 01 dez 2022.

GAMES, R. Matchmaking and Autofill. 2022. Disponível em: <https://support-leagueoflegends.riotgames.com/hc/en-us/articles/201752954-Matchmaking-and-Autofill>. Acesso em: 01 dez 2022.

GRAEPEL, T.; HERBRICH, Ranking and matchmaking. *Game Developer Magazine*, v. 25, p. 34, 2006.

MINKA, T.; CLEVEN, R.; ZAYKOV, Y. *Trueskill 2: An improved bayesian skill rating system*. Technical Report, 2018.

MYŚLAK, M.; DEJA, D. Developing game-structure sensitive matchmaking system for massive-multiplayer online games. In: SPRINGER. *International Conference on Social Informatics*. [S.l.], 2014. p. 200–208.

STEAM. Steam Store. 2022. Disponível em: <<https://store.steampowered.com/>>. Acesso em: 15 dez 2022.

VÉRON, M.; MARIN, O.; MONNET, S. Matchmaking in multi-player on-line games: studying user traces to improve the user experience. In: *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*. Workshop. [S.l.: s.n.], 2014. p. 7–12.