

# Sistema Móvel Organizador de Campeonatos de Esportes Individuais

Carlos E. Pasin<sup>1</sup>,  
Alexandre Perin de Souza<sup>1</sup>

<sup>1</sup>Instituto Federal de Santa Catarina (IFSC)  
Lages – SC – Brasil R. Heitor Villa Lobos, 225 - Sao Francisco, Lages - SC, 88506-400

carlos.eapasin@gmail.com,

alexandre.perin@ifsc.edu.br

**Abstract.** *This article proposes an organizer system for tournaments and challenges, developed to assist players in competing with each other without the need to manually organize the championships. It separates the rankings by sport, enhancing competitiveness by enabling competitions for placement through challenges, that allows users to challenge players who are one position above their position.*

**Resumo.** *Este artigo propõe um sistema organizador de campeonatos e desafios, desenvolvido para auxiliar jogadores a competirem entre si sem a necessidade de organizar manualmente os campeonatos, separando os rankings por esporte, aumentando a competitividade ao possibilitar disputas por colocação através de desafios, que possibilitará que os usuários desafiem os jogadores que estão uma posição acima deles.*

## 1. Introdução:

Os esportes são considerados uma parte integrante da cultura e do estilo de vida das pessoas. Eles também podem ter um impacto significativo na saúde e no bem-estar das pessoas, proporcionando uma oportunidade de melhorar sua aptidão física e mental, como destacado por (Silk, 2016).

Conforme enfatizado por Guttmann (1991), os esportes também se destacam como uma forma de competição organizada entre indivíduos ou equipes. Além de serem uma forma de entretenimento e lazer, os esportes podem ter um papel fundamental na construção de valores como a competitividade e a superação de *desafios*, como defendido por (Coakley, 2011).

Segundo Oliveira (2020), os esportes competitivos geralmente incentivam a criação e gestão de partidas, o que pode ser um *desafio* para os organizadores. De acordo com Almeida (2019), a organização de torneios depende de um esforço manual, o que demanda muito tempo para definir o formato do *campeonato* e pode levar a falhas na organização e na definição das partidas.

Nesse contexto, ao analisar o padel, que é um esporte de raquete que combina elementos do tênis e do squash, nos períodos de ociosidade entre um *campeonato* e outro, os jogadores têm buscado alternativas para manter suas habilidades em dia. Nesse sentido, as disputas entre os jogadores de padel podem ser uma excelente maneira de treinar e

manter-se competitivo. Essas disputas oferecem aos jogadores a oportunidade de testar suas habilidades em situações reais de jogo, identificar pontos fortes e fracos a serem trabalhados, desenvolver a capacidade de tomar decisões rápidas e estratégicas durante a partida enquanto praticam.

O objetivo deste projeto é desenvolver um software que possibilite a organização de disputas e *campeonatos* de esportes, realizando atualizações das colocações das equipes nos respectivos *rankings* (classificação). Este trabalho tem como objetivos específicos:

- Identificar, selecionar e analisar os trabalhos semelhantes para entender como são gerenciados os esportes;
- Projetar e implementar uma aplicação móvel;
- Avaliar a aplicação móvel desenvolvida.

Para a implementação do projeto, foram definidas cinco etapas. Na primeira etapa, será realizada uma análise de trabalhos similares, a fim de compreender como são organizadas partidas e *campeonatos*. Para isso, serão utilizados recursos como o Google Scholar, repositórios institucionais e bases de dados digitais disponíveis na internet. Na segunda etapa, será elaborado o trabalho teórico, que incluirá a introdução do projeto, a apresentação da problemática, dos objetivos e da metodologia. Para isso, será utilizado o software Overleaf. Na terceira etapa, serão criados protótipos do projeto, utilizando softwares de modelagem. Na quarta etapa, será implementada a aplicação móvel do sistema organizador de *campeonatos* e *desafios*, utilizando o Android Studio. Finalmente, na quinta etapa, a aplicação será disponibilizada para avaliação, utilizando formulários do Google Forms para obter *feedbacks* dos usuários.

Do ponto de vista da natureza, este trabalho é classificado como pesquisa aplicada, pois visa aplicar conhecimentos e solucionar problemas práticos em um determinado contexto. Em relação a abordagem do problema é classificada como qualitativa, ou seja, o foco será na compreensão e interpretação dos dados coletados por meio de entrevistas, observações ou análise de documentos. Sobre os objetivos desse projeto são classificados como exploratórios, pois buscam compreender as relações causais entre os fenômenos estudados a partir de um problema. Do ponto de vista dos procedimentos técnicos, esse projeto será realizado por meio de uma pesquisa bibliográfica, que consiste na busca, seleção e análise crítica de informações e dados disponíveis em fontes documentais, tais como livros, artigos científicos, teses e dissertações.

Este artigo está organizado da seguinte maneira. Após a seção introdutória, a seção dois apresenta o referencial teórico, incluindo descrições dos assuntos necessários para o entendimento do trabalho e dos principais trabalhos relacionados com o tema. Em seguida, a seção três registra em detalhes o desenvolvimento do projeto, seguida pela seção quatro que apresenta a avaliação do projeto. Por fim, a seção cinco encerra o documento com as considerações finais.

## **2. Referencial Teórico**

Esta seção está dividida em quatro partes. A primeira parte explica sobre os *campeonatos* e como eles são organizados. A segunda parte contextualiza os *desafios*. A terceira parte Tecnologias e Aplicações. A quarta parte discorre sobre trabalhos com temas relacionados a esportes.

## 2.1. Esportes

Os esportes são atividades físicas competitivas, que são regidas por regras estabelecidas e envolvem uma ampla gama de habilidades físicas e mentais, tais como agilidade, força, velocidade, coordenação, estratégia e tática. Essas habilidades são aplicadas de forma intencional para alcançar um objetivo específico, seja ele vencer uma competição ou melhorar o desempenho pessoal. O esporte é geralmente praticado em um ambiente controlado, com um conjunto de normas e regulamentos que visam garantir a segurança e a justiça para todos os participantes. Além disso, os esportes podem ter uma variedade de benefícios, incluindo a promoção da saúde e da aptidão física, o desenvolvimento de habilidades sociais e emocionais, e a promoção de valores como ética, disciplina e trabalho em equipe (Holowchak, 2012).

Um exemplo de esporte, o padel, é uma atividade física que combina elementos do tênis e do squash e é jogado em uma quadra menor do que a de tênis, chamada de *padel court*. Os jogadores usam raquetes e uma bola específica para o esporte, que é semelhante a uma bola de tênis, mas com menos pressão.

O objetivo do jogo é acertar a bola na quadra do adversário sem que ele consiga devolvê-la antes que ela quique duas vezes. O jogo pode ser disputado em duplas ou individualmente e a pontuação segue o sistema de *vantage set* do tênis, com a diferença de que não existe o *deuce* e o jogador ou dupla que alcançar seis pontos primeiro, com uma diferença de pelo menos dois games, vence o set.

## 2.2. Campeonatos

Um *campeonato* pode ser definido como um conjunto de jogos ou eventos esportivos envolvendo dois ou mais competidores ou equipes, que competem entre si para determinar o vencedor, com base em um conjunto de regras e critérios pré-determinados (Shmanske, 2012).

Os *campeonatos* geralmente são organizados em formato de tabela, chaveamento, com jogos disputados em rodadas ou etapas, e podem ter diferentes níveis de complexidade e duração, desde *campeonatos* locais de curta duração até *campeonatos* internacionais de longa duração.

A modalidade dos *campeonatos* em tabela, também conhecidos como torneios de liga, são organizados em um formato de tabela, onde cada equipe ou participante joga um número pré-determinado de partidas contra todos os outros participantes. A pontuação é atribuída com base no número de vitórias e empates, e o vencedor é determinado pela pontuação acumulada no final da competição. Tais *campeonatos* em tabela são comumente utilizados em modalidades esportivas como futebol, basquete e vôlei.

A modalidade referente a *campeonatos* chaveados, também conhecidos como torneios eliminatórios, são organizados em um formato de chaves, onde cada participante ou equipe é emparelhado e o vencedor avança para a próxima fase, enquanto o perdedor é eliminado, até que reste apenas um vencedor. Os *campeonatos* chaveados são comumente utilizados em modalidades esportivas como tênis.

## 2.3. Desafios

O sistema de *desafios* no padel, tênis e squash por exemplo, permite que os jogadores desafiem outros jogadores que estão uma posição acima no mesmo *ranking*. Por exemplo,

se um jogador estiver na quinta posição do *ranking*, ele pode desafiar o jogador que está em quarto lugar do *ranking*.

Se o jogador que está sendo desafiado aceitar o *desafio*, uma partida é agendada entre os dois jogadores. Caso o desafiante vencer, ele sobe uma posição no *ranking*, e o jogador que perdeu desce uma posição. O jogador que venceu, mantém sua posição no *ranking* e o desafiante permanece na mesma posição.

Esse sistema incentiva os jogadores a jogarem mais partidas e a melhorarem seu desempenho, para que possam subir na classificação e desafiar jogadores mais bem colocados. Além disso, o sistema permite que jogadores de habilidade semelhante joguem entre si e melhorem suas habilidades ao enfrentar jogadores mais experientes.

## 2.4. Trabalhos Relacionados

Nesta subseção são descritos trabalhos que também tem como objetivo organizar *ranking*, *campeonatos* ou *desafios* de esportes.

A pesquisa foi realizada no Google utilizando a string de consulta “organizador de *campeonatos*”. A partir do resultado da consulta foram filtrados 13 (treze) trabalhos e separados para análise e leitura. Observou-se que os trabalhos posteriores a esses 13 tinham pouca ou nenhuma relação com o tema. Para cada um desses trabalhos foi realizada a leitura pelo menos do resumo e foi quantificada sua relevância de forma empírica para o domínio do problema, analisando o quanto cada trabalho se aproximava ou distanciava da temática proposta, utilizando-se uma escala de valores de 1 a 5, em que 1 (um) e pouco relevante e 5 (cinco) muito relevante. Do total de 13 trabalhos foram selecionados os 10 (dez) mais relevantes. Foi realizada uma leitura completa dos trabalhos selecionados e foram levantadas e detalhadas as seguintes informações: quantas vezes foi citado tipo de Trabalho (artigo, monografia, dissertação, tese, etc.), título do trabalho, periódico/conferência, ano de publicação, objetivo, tecnologias, estágio de desenvolvimento, resultados e nível de ensino.

Baseado nesse estudo, no quadro 1 mostra os principais trabalhos relacionados encontrados na literatura. Esses trabalhos diferem em maior parte em três características: sistema de *desafios*, *ranking*, sistema de *campeonato*.

<i>Nome</i>	<i>Criação de competição</i>	<i>Ranking</i>	<i>Desafios</i>	<i>Mobile</i>	<i>Vários esportes</i>	<i>Gratuito</i>
<b>Webcup</b>	x					x
<b>Copa Fácil</b>	x	x		x	x	
<b>Leagues</b>	x	x				x
<b>Ifut</b>	x			x	x	x
<b>Letzplay</b>	x	x		x	x	x
<b>Metafy</b>	x				x	
<b>Tournament Manager</b>				x	x	x
<b>Organizador de torneios</b>	x			x	x	x
<b>Futebol de Varzea</b>	x				x	x
<b>Carlos Pasin</b>	x	x	x	x	x	x

**Quadro 1. Ferramentas Similares**

## 3. Desenvolvimento

Esta seção é dedicada ao desenvolvimento do sistema e está dividida em três partes. Na subseção 3.1 apresenta as ferramentas e tecnologias utilizadas para o desenvolvimento do

sistema. A subseção 3.2 descreve a modelagem do sistema, seus requisitos funcionais e não funcionais, o diagrama de casos de uso, a prototipagem das interfaces e a modelagem do banco de dados. Já a subseção 3.3 aborda a implementação do Sistema Móvel Organizador de Campeonatos de Esportes Individuais e as informações relevantes para o desenvolvimento dos módulos necessários para o funcionamento do sistema.

### **3.1. Ferramentas e Tecnologias**

Nesta subseção, serão apresentadas informações descritivas sobre cada uma das tecnologias e ferramentas que serão utilizadas na construção do desenvolvimento do projeto.

#### **3.1.1. Visual Studio Code**

O Visual Studio Code é mais do que apenas um editor, ele possui muitos recursos que permitem executar com eficiência diversas tarefas relacionadas à programação e desenvolvimento de sistemas. O grupo de software que desenvolve a programação é chamado de ambiente de desenvolvimento. O Visual Studio Code é um dos principais ambientes de desenvolvimento usados por muitas linguagens de programação (Academy, 2020).

O Visual Studio Code terá grande importância para auxiliar na programação e desenvolvimento do sistema. Será utilizado tanto para a codificação do *front-end* como para o *back-end*.

#### **3.1.2. GitHub**

O Git é um tipo de ferramenta de controle de versão de arquivos, que foi criado para gerenciar quem e quando foi editada a versão do código-fonte, qual versão é a mais recente e assim por diante. Um recurso do Git é que o controle de versão, como gravação e rastreamento de código-fonte. É mais fácil para os engenheiros reverter o código-fonte, organizar o histórico de modificações e registrá-lo, tornando-o mais conveniente para os engenheiros (Modis, 2020).

O GitHub é uma plataforma para utilização do Git, e que foi utilizado no desenvolvimento da aplicação deste estudo. Neste trabalho sua função seria criar um histórico de cada etapa que estará sendo desenvolvida, para, se necessário, podermos retroagir a versão caso necessário.

#### **3.1.3. Figma**

O Figma é uma ferramenta online para design vetorial de interfaces e protótipos. Destaca-se por ser gratuito, colaborativo com alterações em tempo real, operada através do próprio navegador, sem a necessidade de instalar um novo software em seu computador. Ele será essencial para a prototipação da interface gráfica da aplicação, sendo possível ter uma noção mais precisa de como ficará o projeto das interfaces do *front-end*.

### 3.1.4. ReactNative

O React Native é um framework de desenvolvimento de aplicativos móveis criado pelo Facebook que permite o desenvolvimento de aplicativos nativos para iOS e Android usando JavaScript e React, uma biblioteca JavaScript para construção de interfaces de usuário. Ele é uma extensão do popular framework React, que é usado para desenvolvimento web. Ela será a ferramenta principal para o desenvolvimento do *front-end* da aplicação.

### 3.1.5. Expo

O Expo é uma biblioteca JavaScript amplamente usada no desenvolvimento de aplicativos móveis para dispositivos iOS e Android, popular por simplificar o desenvolvimento e pela possibilidade de criar aplicativos de forma rápida e direta, sem a necessidade de lidar com muitos detalhes técnicos. A biblioteca facilitará na validação de cada alteração na aplicação, atualizando em tempo real a cada alteração salva e também no deploy da aplicação após a finalização do projeto.

### 3.1.6. Django

Django é um *framework web Python* de alto nível que incentiva o desenvolvimento rápido e um design limpo e pragmático DjangoWebsite (2018), que permite criar aplicações *web* com muitas facilidades. É gratuito e de código aberto, tem uma comunidade próspera e ativa, ótima documentação e muitas opções de suporte gratuito e pago (MozillaContributors, 2022). Será utilizada para o desenvolvimento do *back-end* da aplicação, onde facilitará na criação das *APIs* e do *Banco de dados*.

### 3.1.7. Django Rest Framework

Django REST *Framework* é uma biblioteca que permite a construção de APIs REST utilizando a estrutura do Django, permitindo a construção de APIs em qualquer sistema operacional. É muito utilizado por toda a comunidade, pois provê uma forma simples e rápida para a construção de APIs utilizando recursos o sistema de rotas e seu mapeamento objeto-relacional para manipulação de banco de dados (de Andrade, 2020). Auxiliará na criação e gerenciamento das APIs, facilitando todo o processo.

### 3.1.8. SQLite

SQLite é uma biblioteca de linguagem C que implementa um mecanismo de banco de dados SQL pequeno, rápido, independente, de alta confiabilidade e completo SQLiteWebsite (2022). Funciona como um servidor próprio e independente, já que o SGBD (Sistema de Gerenciamento de Banco de Dados) pode ser executado na mesma instância, o que elimina consultas e processos separados. Portanto, a biblioteca SQLite é gerada e armazenada diretamente no arquivo do banco de dados (Vieira, 2022). Será utilizado como o banco de dados principal da aplicação, atuando juntamente com o *back-end* e as *APIs*.

### 3.2. Protótipo das Interfaces

Através do software Figma, foram desenvolvidos protótipos iniciais que serviram como base para o desenvolvimento e evolução da aplicação, definindo como seriam desenvolvidas as interfaces, a estrutura do projeto e componentes, facilitando a visualização do sistema como um todo. Serão apresentados nesta subseção, as interfaces de tela inicial, ranqueamento e criação de *campeonatos*, que seriam as telas principais para ter como base para o desenvolvimento das demais telas e por ser as mais utilizadas pelos usuários.



**Figura 1. Tela inicial**

O protótipo, conforme apresentado na Imagem 1, apresenta a tela inicial do sistema, onde apresentará as opções de entrada do aplicativo, já introduzindo as cores padrões do sistema e um protótipo de logo.



**Figura 2. Tela de Ranking**

Na imagem 2, apresenta a tela de *ranking*, onde será apresentado o *rankig* de cada esporte para o *usuário* acessar e conseguir validar as colocações dos jogadores.



**Figura 3. Tela de Criação de campeonato**

Na imagem 3, apresenta a criação de *campeonatos*, onde apresentará as opções de esportes disponíveis e os *usuários* adicionados no sistema para participar da competição. Assim realizando a adição dos participantes em times de duplas ou mantendo individual, realizando assim as partidas de forma aleatória e atualizando a pontuação dos *usuários* nos *rankings* conforme o andamento dos *campeonatos*.

### **3.3. Diagramas de caso de usos**

Para uma mais clara compreensão dos requisitos da aplicação proposta, foi desenvolvido o diagrama de casos de uso da *Unified Modeling Language* (UML). O diagrama apresentado na imagem 4 mostra quais funcionalidades poderão ser acessadas pelo *usuário*.

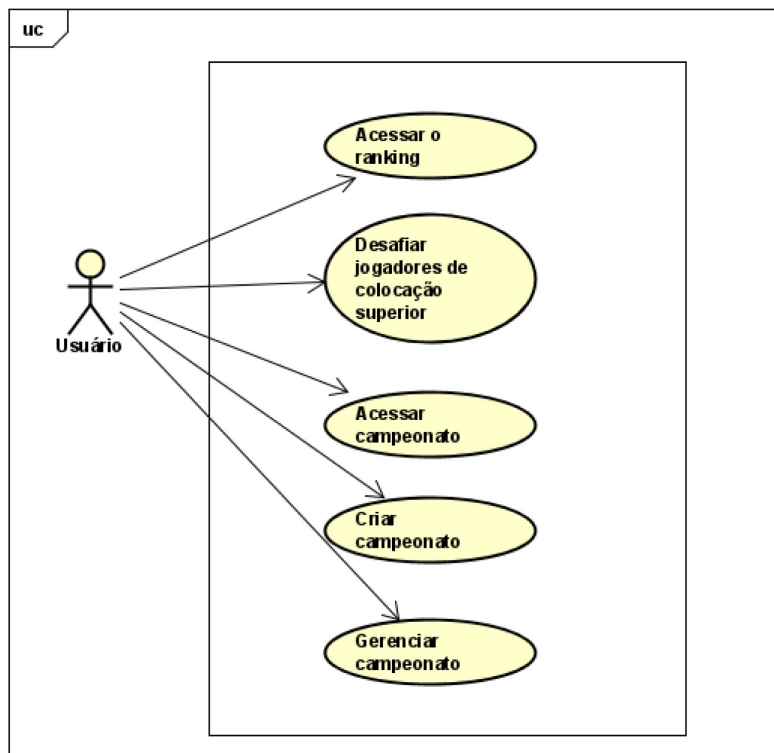


Figura 4. Casos de uso: Usuário

O *Usuário* tem acesso a todas as telas e opções disponíveis no sistema, apenas não podendo gerenciar *campeonatos* que não tem permissão.

### 3.4. Modelo de banco de dados

A modelagem do banco de dados foi feita a partir de requisitos e das interfaces prototipadas.

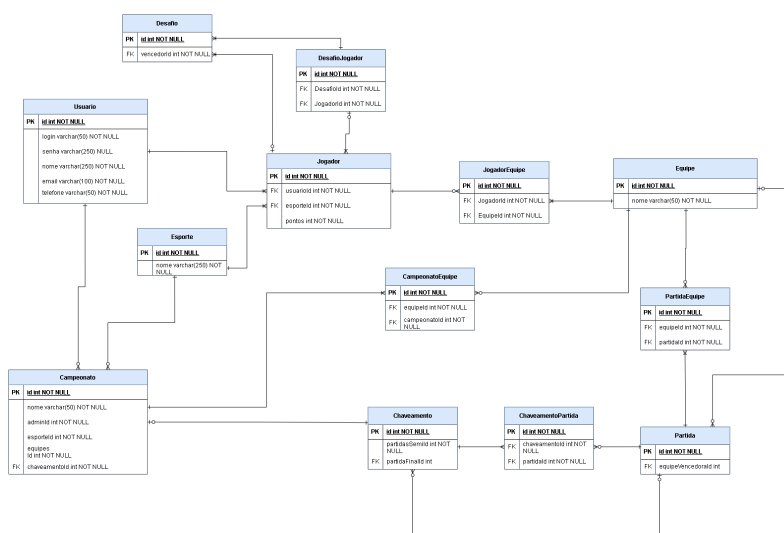
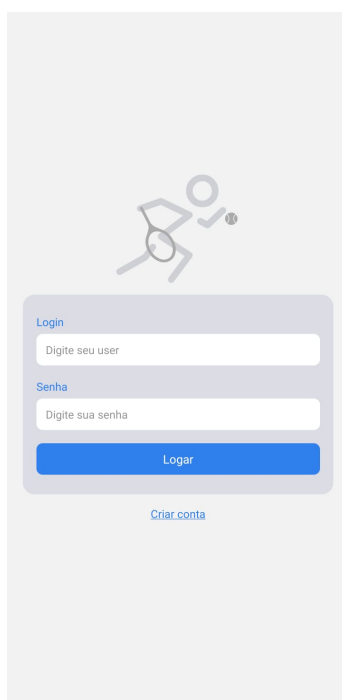


Figura 5. Banco de dados

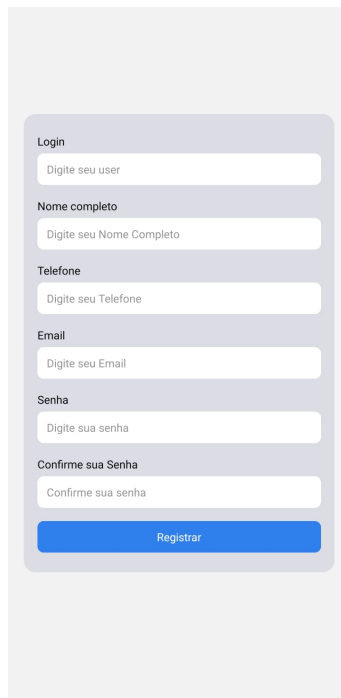
### 3.5. Desenvolvimento do Front-end

Das imagens 6 a 19 apresentam as telas que foram desenvolvidas dentro deste projeto utilizando as tecnologias e bibliotecas *React Native* e *Expo* que auxiliaram a criação mais simples das interfaces, possibilitando visualizar as alterações em tempo real e facilitando na publicação do app após completar o desenvolvimento.

Com base nos protótipos apresentados anteriormente, foram criadas as seguintes interfaces, mantendo um padrão no aplicativo e mantendo-o intuitivo.



**Figura 6. Tela de Login**



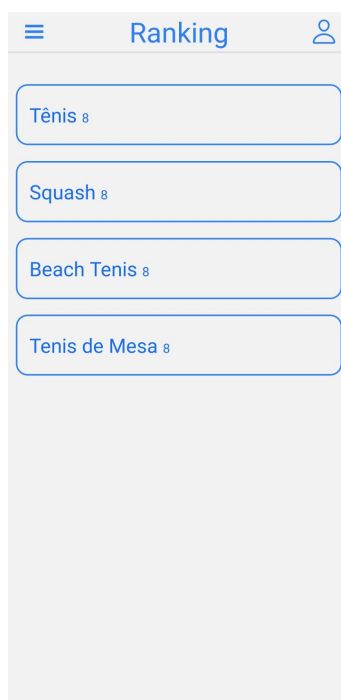
A tela de cadastro apresenta um formulário com os seguintes campos:

- Login**: Digite seu user
- Nome completo**: Digite seu Nome Completo
- Telefone**: Digite seu Telefone
- Email**: Digite seu Email
- Senha**: Digite sua senha
- Confirme sua Senha**: Confirme sua senha

Um botão azul "Registrar" está localizado na base do formulário.

**Figura 7. Tela de Cadastro**

Nas imagens 6 e 7, apresenta a tela de Login e de Registro do sistema, o *usuário* poderá realizar o login e acessar o sistema, ou realizar um novo registro criando uma conta, onde preencherá os seus dados e informará o seu login e senha para acessar o sistema.

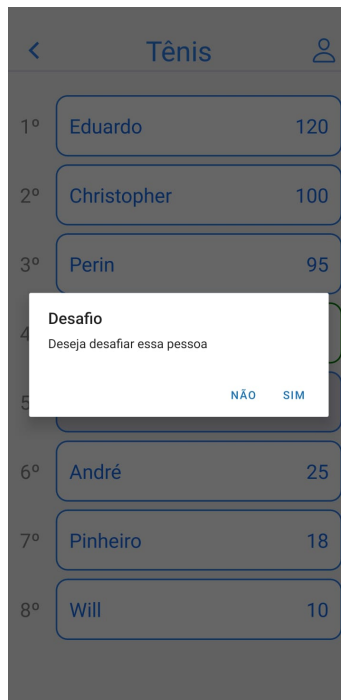


**Figura 8. Tela de Ranking - Esportes**

Na imagem 8, é apresentado o *ranking* dos esportes cadastrados com registro de pontuação de *usuários*, podendo acessar os *rankings* de cada esporte para visualização dos jogadores e suas pontuações, podendo acessar também o seu perfil pressionando o ícone de usuário, assim visualizando os seus dados e podendo editá-los. Ao pressionar o ícone de hambúrguer, o *usuário* poderá acessar a tela de *campeonato* e sair da conta que está logada, fazendo-o retornar para tela inicial.

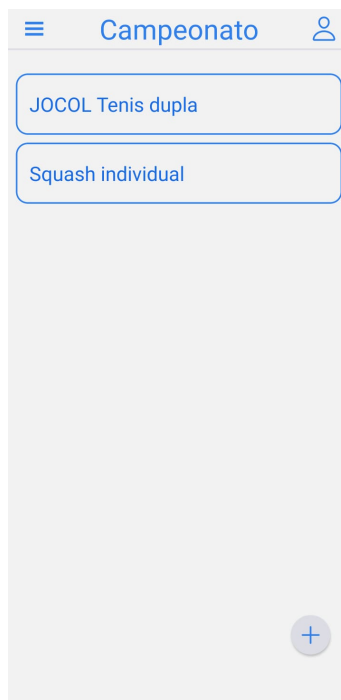
Rank	Player Name	Score
1º	Eduardo	120
2º	Christopher	100
3º	Perin	95
4º	Joca	90
5º	Carlos (Você)	75
6º	André	25
7º	Pinheiro	18
8º	Will	10

**Figura 9. Tela de Ranking - Jogadores**



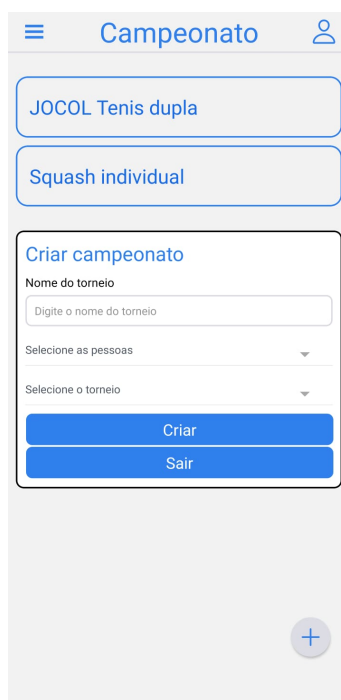
**Figura 10. Tela de Ranking - Desafio**

Na imagem 9, é apresentada a tela de *ranking* de jogadores dentro do esporte Tênis, informando o nome e pontuação de todos os jogadores que possuem uma pontuação no esporte. Identificando ao *usuário* em que posição ele se encontra no *ranking* e informando sempre que o jogador acima de sua posição está disponível para ser *desafiado*, como apresentado na imagem 10, possibilitando realizar o *desafio*, caso o oponente aceitar. Ocasionalmente na troca de pontos caso o desafiante vencer do desafiado.



**Figura 11. Tela de Campeonato**

Na imagem 11, é apresentada a tela de *campeonato*, onde é possível realizar o cadastro de novos *campeonatos*, visualizar os *campeonatos* em andamentos, acessar o seu perfil pressionando o ícone de *usuário*, assim visualizando os seus dados e podendo editá-los. Ao pressionar o ícone de hambúrguer, o *usuário* poderá acessar a tela de *ranking* e sair da conta que está logada, fazendo-o retornar para tela inicial.



**Figura 12. Tela de Criação de Campeonato**

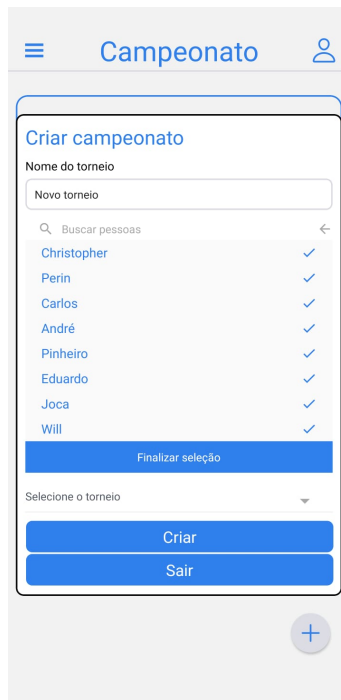


Figura 13. Tela de Criação de Campeonato - Seleção de Jogadores

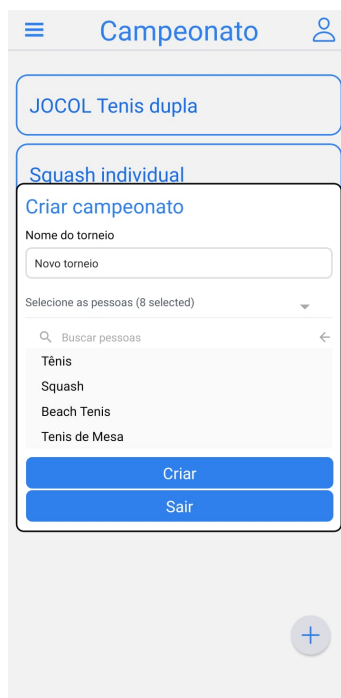
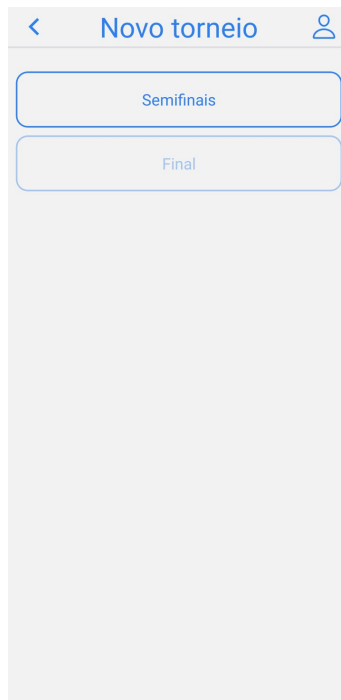


Figura 14. Tela de Criação de Campeonato - Seleção de Esporte

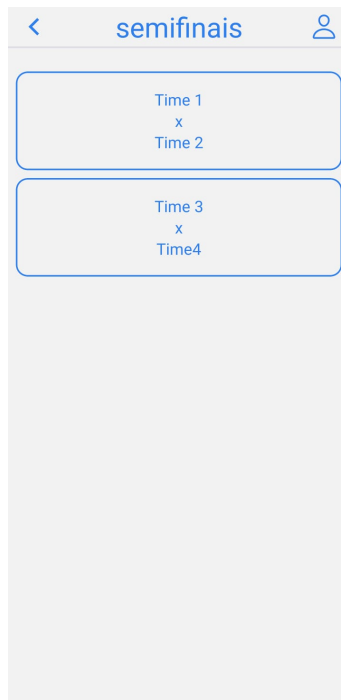


**Figura 15. Tela de Campeonato - Campeonato Criado**

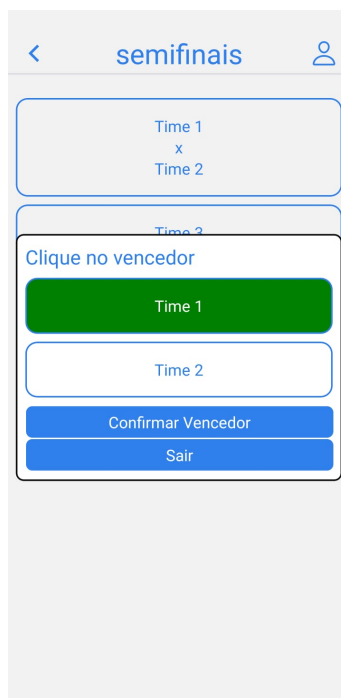
Nas imagens 12, 14, é apresentado a tela de criação de *campeonato*, onde é possível informar um nome para o *campeonato*, selecionar jogadores cadastrados no sistema, selecionar esportes cadastrados no sistema e criar. Apresentando assim a imagem 15, onde apresenta as fases do *campeonato* e em qual fase o *campeonato* está.



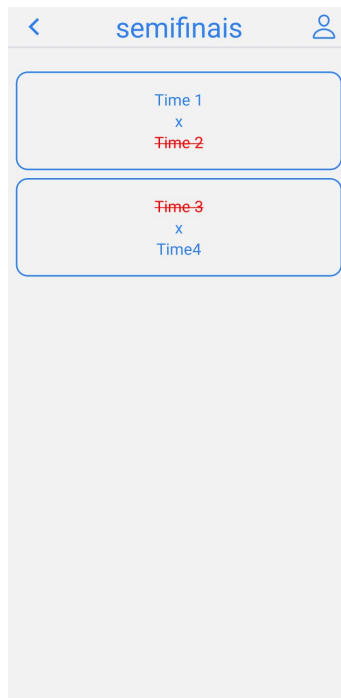
**Figura 16. Tela de Campeonato - Partidas Jogador X Jogador**



**Figura 17. Tela de Campeonato - Partidas Time X Time**



**Figura 18. Tela de Campeonato - Selecionar o Vencedor**



**Figura 19. Tela de Campeonato - Apresentação dos Perdedores**

Nas imagens 16 e 17, são apresentadas como são organizados os confronto por times de duplas e individual, apresentando uma tela para confirmar o vencedor ao selecionar a partida, como apresentado na imagem 18, assim adicionando 3 pontos para o vencedor caso seja uma partida comum e 5 pontos caso seja uma *final*. Marcando o perdedor alterando o nome para vermelho e eliminando-o do *campeonato*, assim liberando o confronto dos times vencedores para a próxima etapa.

### **3.6. Desenvolvimento do back-end**

Após a criação das interfaces gráficas e a validação das mesmas, partiu-se para o desenvolvimento do back-end da aplicação. O primeiro passo, foi a criação do projeto através da utilização do *Django* e do *Django Rest Framework*, que cria como padrão um projeto com uma arquitetura (*Model-View-Template*) MVT. Porém, para esse projeto, utilizou-se apenas a parte de modelos e *views* para a criação de uma API, para comunicar com o *front-end* feito em um projeto separado. De acordo com o padrão, foram criados três arquivos: *models*, *views* e *urls*. Eles são os principais arquivos que foram modificados para o desenvolvimento da API, pois neles são armazenados os dados sobre a aplicação no *back-end*. Também foi criado um arquivo para realizar a conexão com o banco de dados SQLite. O arquivo *models.py* contém as informações sobre os objetos e suas relações, o arquivo *views.py* contém as chamadas de *GET*, *POST*, *PUT* e *DELETE* de cada um dos objetos, e por fim, o arquivo *urls.py* é onde são definidas as rotas por onde os recursos disponibilizados pela API possam ser acessados.

A escolha do *Django Rest Framework* no projeto, se deu por duas razões importantes. A primeira razão reside no fato da separação do *back-end* e *front-end* para que pudessem serem desenvolvidas de forma independentes. A segunda razão é pela facilidade do *framework*, primeiro na criação de banco de dados, que cria toda a estrutura do

banco a partir de como são construídos os modelos em código, e segundo na criação das chamadas básicas de *CRUD* de cada dos objetos.

A API criada recebe e envia dados através de documentos *json* para suas requisições, seja para receber dados que irão ser usados para alterações dentro do sistema, quanto para enviar informações de dados dos objetos. Isso é usado também pelo *front-end* que utiliza os arquivos de resposta *json* para mostrar as informações vindas no *back-end*. No arquivo *views.py*, foram criadas funções *CRUD* para todos os objetos, porém, foram feitas algumas modificações para adição de parâmetros na resposta *json*. Um exemplo disso é mostrado nas imagens 20, 21 e 23. Cada uma das rotas em *url* são definidas em um arquivo *urls.py*, onde concentram as rotas de todas as chamadas da API, conforme é mostrado na imagem 23. Isto, permite uma melhor organização do código e uma mais fácil manutenção e evolução do *back-end*. 18

```
class ChallengeViewSet(viewsets.ModelViewSet):
    queryset = Challenge.objects.all()
    serializer_class = ChallengeSerializer

    @action(detail=False, methods=['get'])
    def getChallenges(self, request):
        challenges = Challenge.objects.all()
        response = ChallengeSerializer(challenges, many=True)
        return Response({
            'success' : True,
            'challenges' : response.data})

    @action(detail=True, methods=['get'])
    def getChallenge(self, request, pk):
        challenge = Challenge.objects.filter(id=pk)
        response = ChallengeSerializer(challenge, many=True)
        return Response({
            'success' : True,
            'challenge' : response.data})

    @action(detail=True, methods=['put'])
    def ExchangePlayersPoints(self, request, pk):
        challenge = Challenge.objects.filter(id=pk).first()
        player1 = challenge.players.first()
        player2 = challenge.players.last()
        winnerId = challenge.winner.pk
        newPoints = 1

        if(player1.id != winnerId):
            newPoints = player1.points
            player1.points = player2.points
            player2.points = newPoints
        else:
            newPoints = player2.points
            player2.points = player1.points
            player1.points = newPoints

        player1.save()
        player2.save()

        return Response({'success' : True})
```

Figura 20. Exemplo código de chamada *GET*

```

class TeamViewSet(viewsets.ModelViewSet):
    queryset = Team.objects.all()
    serializer_class = TeamSerializer

    @action(detail=False, methods=['get'])
    def getTeams(self, request):
        team = Team.objects.all()
        response = TeamSerializer(team, many=True)
        return Response({
            'success' : True,
            'team' : response.data})

    @action(detail=True, methods=['get'])
    def getTeam(self, request, pk):
        team = Team.objects.filter(id=pk)
        response = TeamSerializer(team, many=True)
        return Response({
            'success' : True,
            'team' : response.data})

class MatchViewSet(viewsets.ModelViewSet):
    queryset = Match.objects.all()
    serializer_class = MatchSerializer

    @action(detail=False, methods=['get'])
    def getMatches(self, request):
        matches = Match.objects.all()
        response = MatchSerializer(matches, many=True)
        return Response({
            'success' : True,
            'matches' : response.data})

    @action(detail=True, methods=['get'])
    def getMatch(self, request, pk):
        match = Match.objects.filter(id=pk)
        response = MatchSerializer(match, many=True)
        return Response({
            'success' : True,
            'match' : response.data})

```

Figura 21. Exemplo código de chamada GET'

```

    @action(detail=True, methods=['put'])
    def addThreePoints(self, request, pk):
        player = Player.objects.filter(id=pk).first()
        player.points += 3
        player.save()
        response = PlayerSerializer(player, many=True)
        return Response({
            'success' : True,
            'player' : response.data})

    @action(detail=True, methods=['put'])
    def addFivePoints(self, request, pk):
        player = Player.objects.filter(id=pk).first()
        player.points += 5
        player.save()
        response = PlayerSerializer(player, many=True)
        return Response({
            'success' : True,
            'player' : response.data})

```

Figura 22. Exemplo código de chamada GET'

```

router = routers.DefaultRouter()
router.register('user', views.UserViewSet)
router.register('player', views.PlayerViewSet)
router.register('sport', views.SportViewSet)
router.register('team', views.TeamViewSet)
router.register('match', views.MatchViewSet)
router.register('championship', views.ChampionshipViewSet)
router.register('challenge', views.ChallengeViewSet)
router.register('bracket', views.BracketViewSet)

urlpatterns = [
    path('', include(router.urls)),
    path('index', views.index)
]

```

Figura 23. Exemplo código de rotas

#### 4. Avaliação da Aplicação

Esta seção aborda a avaliação da aplicação. Para conduzir essa avaliação, desenvolvemos um formulário utilizando o *Google Forms*, escolhido pela sua facilidade de uso e pela conveniência de permitir o envio pela internet dos formulários preenchidos. A aplicação foi submetida à avaliação de usuários, quais participam regularmente de *campeonatos* e já possuem familiaridade com o funcionamento de *rankings* e competições esportivas. As perguntas formuladas no questionário visaram obter insights valiosos sobre a experiência do usuário e seu feedback sobre a eficácia da aplicação. As perguntas propostas pelo questionário são:

1. O sistema está adequado (atende) às suas necessidades?;
2. O sistema possui características (interfaces intuitivas, vocabulário conhecido pelo usuário etc) que permitem que ele seja aprendido com facilidade?;
3. O sistema é atrativo, ou seja, as interfaces estão bem projetadas, os componentes de menu são fáceis de localizar e as cores são adequadas?.
4. As funcionalidades que você acessou e utilizou funcionaram de maneira correta?;
5. Os rankings e campeonatos são apresentados de forma organizada, que permite a visualização e acesso de maneira fácil?
6. Os desafios, a criação e gerenciamento de campeonatos, são intuitivos, obtém facilmente o entendimento ao visualizar?
7. Informe o que você mudaria ou aquilo que você não gostou no sistema.
8. Informe o que você gostou no sistema.
9. Qual nota geral você daria para o sistema?

As primeiras 5 perguntas possuem uma escala com cinco alternativas e cada alternativa possui um valor associado à mesma. As alternativas são:

1. Concordo totalmente;
2. Concordo;
3. Não sei responder;

4. Discordo;
5. Discordo totalmente;

Já as perguntas 7 e 8, são perguntas abertas, onde o avaliador da aplicação tem a oportunidade de escrever sobre o que o agradou e o que o desagradou na aplicação. A última pergunta questiona qual nota o avaliador dá ao sistema, utilizando uma escala de um até dez.

#### 4.1. Análise de resultados

Nesta subsecção, são analisados os resultados sobre cada uma das questões pertencentes ao questionário de avaliação e as conclusões a serem feitas diante das respostas dos entrevistados.

O questionário foi respondido por treze pessoas. A primeira pergunta foi sobre se o sistema atende às necessidades dos usuários para gerenciar campeonatos. A imagem 24 mostra os resultados obtidos e que todos os entrevistados concordam que o sistema é capaz de ser usado para tal fim.

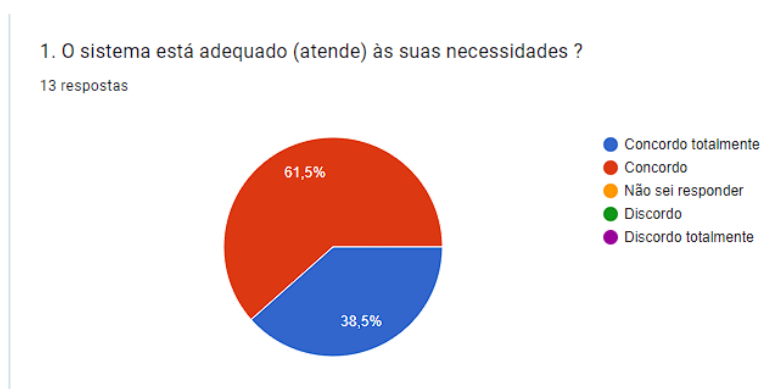
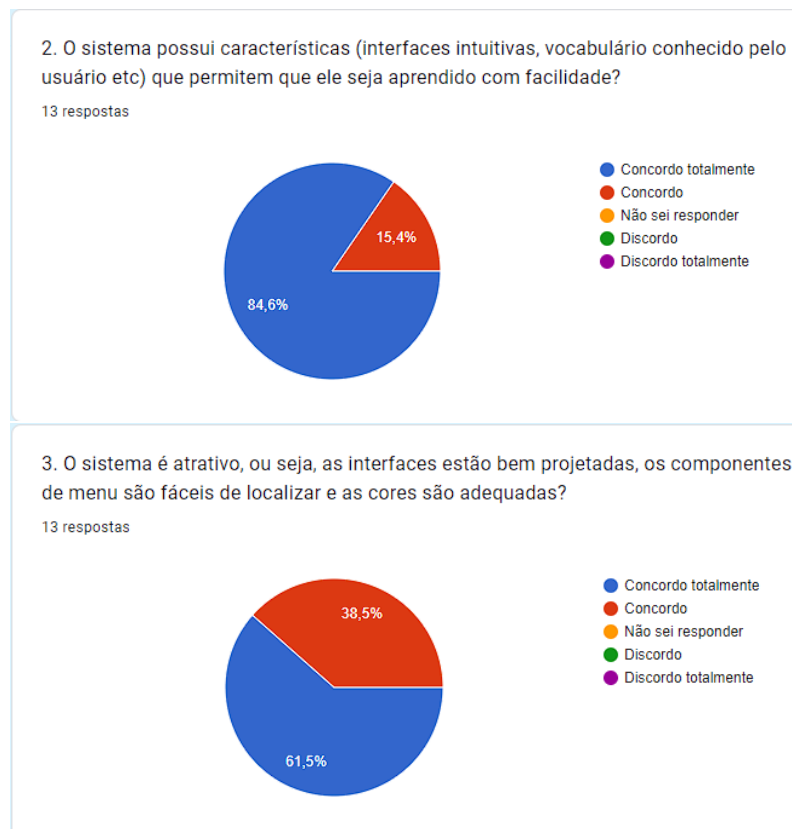


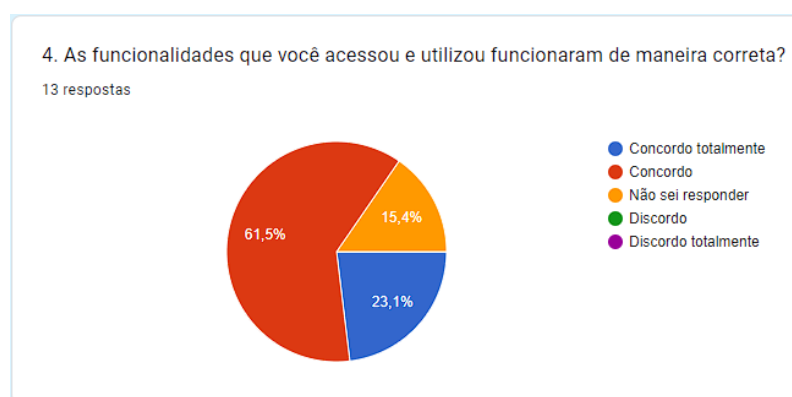
Figura 24. Respostas entrevistados à primeira pergunta do questionário.

A segunda e a terceira pergunta tem o objetivo de verificar se o sistema é intuitivo e atrativo para o usuário e se é fácil de utilizar o sistema diante de como foi feito seu *design*. A imagem 25 mostra o gráfico com as respostas e que os usuários concordam que o sistema é atrativo e possui características que permitem que ele seja de fácil usabilidade.



**Figura 25. Respostas entrevistados à segunda e terceira perguntas do questionário.**

A quarta pergunta procura saber se as funcionalidades do sistema funcionam de maneira correta. A imagem 26 mostra os resultados dessa questão e demonstra que oitenta e quatro por cento dos entrevistados responderam que concordam com a pergunta.



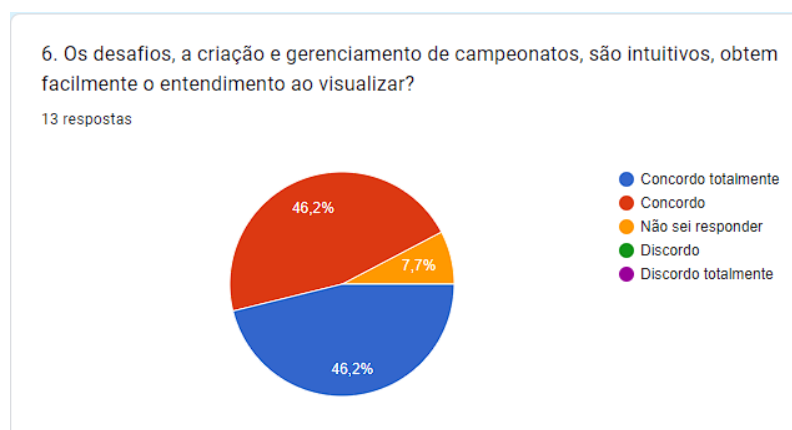
**Figura 26. Respostas entrevistados quarta pergunta do questionário.**

A quinta pergunta é referente aos *rankings* e *campeonatos*, se as informações apresentadas de forma organizada que permite a visualização e acesso de maneira fácil. Segundo a imagem 27, noventa e dois por cento dos usuários concordam que os *rankings* e *campeonatos* atendem às suas necessidades de maneira fácil.



**Figura 27. Respostas entrevistados quinta pergunta do questionário.**

A sexta pergunta, é referente aos *desafios* e a criação e gerenciamento de *campeonatos*, se são apresentadas de forma organizada, que permite a visualização e acesso de maneira fácil. Segundo a imagem 28, noventa e dois por cento dos usuários concordam que os *desafios* e criação e gerenciamento de *campeonato* atendem às suas necessidades de maneira fácil.



**Figura 28. Respostas dos entrevistados para a sexta pergunta do questionário.**

A sétima e oitava perguntas, são questões discursivas quanto ao que os entrevistados gostaram, mudariam ou não gostaram do sistema. Quanto ao que os usuários não gostaram ou mudariam, são destacados o que segue:

- Gostaria de poder ver o perfil de outros membros;
- O perfil pede muitas informações do usuário, poderia ser mais simples ou opcional esta parte;
- Uso de elementos visuais diferentes dentro do que já está implementado no app, pode torná-lo mais atrativo visualmente e mais intuitivo ao acessar diferentes telas/visões com diferentes finalidades (ex.: trazer mais cores, alguns botões um pouco diferentes etc);

Quanto ao que os usuários gostaram no sistema, suas principais considerações foram sobre o *design*, a praticidade e facilidade e utilização do sistema mesmo com pouco uso e também pela ideia de *desafio* dentro dos *rankings* apresentado no projeto.

A última pergunta procura saber em uma escala de zero até dez, uma nota geral que os participantes do questionário dariam à aplicação. Segundo a imagem 29, todos os participantes deram uma nota sete ou maior ao sistema, o que demonstra a satisfação quanto à aplicação e seu uso.

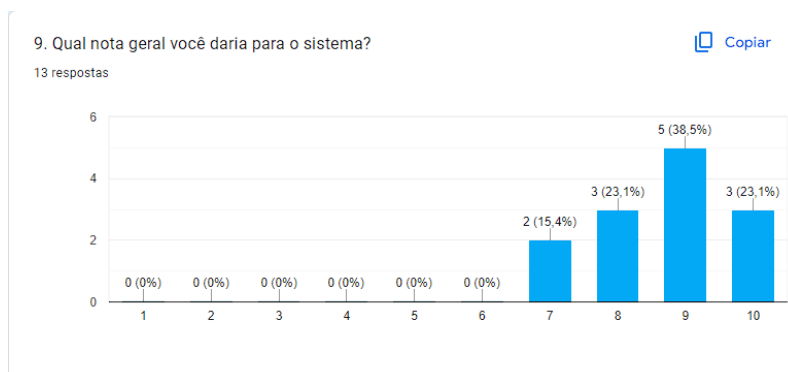


Figura 29. Respostas dos entrevistados para a nona pergunta do questionário.

Veja o apêndice A para ter acesso à todas as respostas do formulário.

## 5. Conclusões

Este trabalho teve como objetivo central o desenvolvimento e implementação de um sistema dedicado à organização de *campeonatos*, *rankings* e *desafios* por posições, com ênfase na experiência do usuário por meio de uma aplicação mobile. Os objetivos de facilitar o gerenciamento de partidas, evitando possíveis erros e agilizando o processo como um todo foram cumpridos. Porém, tiveram funções que não foram possíveis ser realizadas neste projeto, como a possibilidade de cadastrar uma quantidade ilimitada de participantes dentro de um *campeonato* e a criação de times com jogadores atribuídos a ele, por limitação de entendimento na tecnologia utilizada e o tempo estimado para Finalização do sistema. Por conta disso, tivemos que limitar os esportes para apenas esportes individuais, para facilitar e evitar erros na pontuação dos *rankings* e na criação dos *campeonatos*, sendo incluindo inicialmente apenas esportes de raquete.

Os aspectos mais relevantes identificados dentro do sistema, seria a facilidade de uso, onde não é necessário um manual de uso para entender o que o sistema faz, sendo bem intuitivo em todas as telas e ações. A partir da avaliação realizada com a aplicação, conforme está na seção 4, os resultados dos questionários foram satisfatórios, uma vez que a maioria dos avaliadores se mostrou satisfeito com a utilização do sistema. A aplicação funciona e atende aos objetivos estabelecidos, porém, dois aspectos importantes merecem ser destacados para serem desenvolvidos em atividades futuras, sendo eles:

- Interação com outros usuário dentro do aplicativo, seja gerando notificação por email ou até uma interação por chat dentro do sistema.
- Aumentar o limite de jogadores dentro dos *campeonatos*.

Além dessas funcionalidades, faltou a implementação de uma tela de acesso ao perfil do próprio usuário, para atualizar os próprios dados e também a implementação da possibilidade de acessar os dados de outros usuários.

Na realização deste projeto, houve várias dificuldades, não apenas técnicas, mas também lógicas de como o sistema funcionaria. Ainda não tinha conhecimento suficiente para realizar o projeto, porém com muita pesquisa, estudo e dedicação, consegui implementar todas as ações prioritárias planejadas para o funcionamento do sistema. Obtive a ajuda de alguns colegas, especialmente do orientador para entender e resolver muitos problemas encontrados ao longo do projeto.

Seguimos trabalhando em cima das funções que não foram resolvidas, como a atribuição de jogadores dentro de um time e permitir mais jogadores dentro de um *campeonato*. Como atividades futuras, está planejado para desenvolver o acesso ao perfil de outros usuários. Realizar alterações na interface, implementando botões com diferentes cores, mais contraste dentro do aplicativo, para manter um visual mais amigável para os usuários. Implementar um módulo onde habilite a visualizar a organização dos campeonatos de forma gráfica. Gerenciar as permissões dos usuários, ter usuários administradores do sistema e administradores de campeonatos e repassar novamente para mais usuários validarem o sistema.

## Referências

- Academy, T. (2020). Visual studio codevscode. Disponível em: <https://techacademy.jp/magazine/39548>, Acesso em: 23 jun. 2022.
- Almeida, B. S., M. E. L. S. M. R. (2019). *Esportes Complementares*. InterSaberes.
- Coakley, J. (2011). *Youth Sports: What Counts as "Positive Development?"*. Sage Publications.
- de Andrade, A. P. (2020). O que é o django rest framework? Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-o-django-rest-framework>, Acesso em: 12 nov. 2022.
- DjangoWebsite (2018). Django - overview. Disponível em: <https://www.djangoproject.com/start/overview/>, Acesso em: 12 nov. 2022.
- Guttman, A. (1991). Women's sport.
- Modis (2020). Github. Disponível em: [https://www.modis.co.jp/candidate/insight/column\\_30](https://www.modis.co.jp/candidate/insight/column_30), Acesso em: 23 jun. 2022.
- MozillaContributors (2022). Introdução ao django. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Django/Introduction>, Acesso em: 12 nov. 2022.
- Oliveira, E. R., N. C. O. (2020). *Os Novos Desafios da Educação a Distância no Brasil*.
- Silk, M. L., A. D. L. e. T. H. (2016). *Routledge Handbook of Physical Cultural Studies*. Routledge.
- SQLiteWebsite (2022). What is sqlite? Disponível em: <https://www.sqlite.org/index.html>, Acesso em: 12 nov. 2022.
- Vieira, D. (2022). Sqlite: o que é, como funciona e qual é a diferença entre o mysql. Disponível em: <https://www.hostgator.com.br/blog/sqlite-o-que-e-como-funciona-e-qual-e-a-diferenca-entre-o-mysql/>, Acesso em: 12 nov. 2022.

## **A. Apêndice 1**

Link para documento com todas as respostas recebidas na avaliação do *software* : <https://docs.google.com/forms/d/1InQxtYvFA17HsjCkooxfzoHwEhwqrPNaSrziVvoc-uQ/edit?ts=654a840d&pli=1#responses>