



**CENTRO UNIVERSITÁRIO DO DISTRITO FEDERAL – UDF  
COORDENAÇÃO DO CURSO DE SISTEMAS DE INFORMAÇÃO**

**ERICK ARAUJO DE ALENCAR  
LARISSA NOVAIS SARMENTO  
RAFAEL CAETANO DA SILVA**

**RESENHA ESPORTE CLUBE:  
APLICAÇÃO WEB PARA ORGANIZAÇÃO DE PARTIDAS ESPORTIVAS**

**BRASÍLIA  
2024**

**ERICK ARAUJO DE ALENCAR  
LARISSA NOVAIS SARMENTO  
RAFAEL CAETANO DA SILVA**

**RESENHA ESPORTE CLUBE:  
APLICAÇÃO WEB PARA ORGANIZAÇÃO DE PARTIDAS ESPORTIVAS**

Trabalho de conclusão de curso apresentado à  
Coordenação de Sistemas de Informação, do Centro  
Universitário do Distrito Federal - UDF, como requisito  
parcial para obtenção do grau de bacharel em Sistemas  
de Informação.

Orientador: Prof. Me. Gleidson Porto Batista

**BRASÍLIA  
2024**

**IDENTIFICAÇÃO DO DOCUMENTO**

<b>Nome do Projeto:</b> Resenha Esporte Clube: Aplicação Web para Organização de Partidas Esportivas.	
<b>Aluno:</b> Erick Araujo de Alencar, Larissa Novais Sarmento e Rafael Caetano da Silva.	
<b>Orientador:</b> Me.Gleidson Porto Batista.	<b>Data:</b> 02 / 12 / 2024
<b>Arquivo/versão:</b> 1.0	

**ERICK ARAUJO DE ALENCAR  
LARISSA NOVAIS SARMENTO  
RAFAEL CAETANO DA SILVA**

Resenha Esporte Clube

Trabalho de conclusão de curso  
apresentado à Coordenação de Sistemas  
de Informação, do Centro Universitário do  
Distrito Federal - UDF, como requisito  
parcial para obtenção do grau de bacharel  
em Sistemas de Informação.

Orientador: Prof. Me. Gleidson Porto Batista

Brasília, 2 de dezembro de 2024.

Banca Examinadora

---

**GLEIDSON PORTO BATISTA**  
Professor(a) Orientador(a): Mestre  
Centro Universitário - UDF

---

**ELIEL SILVA DA CRUZ**  
Professor(a) Examinador(a): Mestre  
Centro Universitário - UDF

---

**ALEX CASAÑAS**  
Professor(a) Examinador(a): Mestre  
Centro Universitário - UDF

NOTA: 9,5.

Dedico este trabalho à minha família e aos meus amigos,  
cuja presença e apoio foram essenciais para a  
concretização deste projeto.

## **AGRADECIMENTOS**

Gostaríamos de expressar nossa sincera gratidão, primeiramente a Deus, por nos conceder mais um ciclo de aprendizado e crescimento. Agradecemos também ao nosso orientador, professor Gleidson Porto Batista, e à professora Kadidja Valeria Reginaldo de Oliveira, por sua orientação e pelas correções necessárias para melhoria deste trabalho. Estendemos nossa gratidão aos nossos amigos e familiares, que nos apoiaram e estiveram ao nosso lado durante todo esse processo tão especial.

*“O aprendizado é a única coisa que a mente nunca se cansa, nunca tem medo e nunca se arrepende.”*

Immanuel Kant

## RESUMO

Este trabalho apresenta o desenvolvimento de uma aplicação web direcionada à organização de partidas esportivas a fim de auxiliar na inclusão de novos hábitos na rotina da população do Distrito Federal por meio da prática de atividades físicas. A plataforma oferece ao jogador, usuário do sistema, diversas funcionalidades, como a possibilidade da criação de partidas e o gerenciamento delas de forma eficiente. Nesse sentido, todas as etapas para a construção da solução foram rigorosamente seguidas, desde a idealização do projeto até a codificação do sistema. Além disso, foi realizada uma análise dos sistemas disponíveis no mercado, assim como uma pesquisa para mapear os possíveis usuários do sistema. Com base nesse levantamento, algumas dificuldades foram identificadas sobre o desafio de encontrar uma aplicação que centralizasse partidas esportivas. Diante desses aspectos, a necessidade de ferramentas eficazes e ágeis para facilitar a organização de eventos esportivos tornou-se evidente, sendo esse o principal motivo para a criação da aplicação web.

**Palavras-chave:** Esportes, Prática, Projeto, Organização, Partidas.



## **ABSTRACT**

This work presents the development of a web application aimed at organizing sports matches in order to assist in the inclusion of new habits in the routine of the population of the Federal District through the practice of physical activities. The platform offers the player, the system user, several functionalities, such as the possibility of creating matches and managing them efficiently. In this sense, all steps for building the solution were rigorously followed, from project creation to system coding. Furthermore, an analysis of the systems available on the market was carried out, as well as a survey to map the possible users of the system. Based on this survey, some difficulties were identified regarding the challenge of finding an application that centralizes sports matches. Given these aspects, the need for effective and agile tools to facilitate the organization of sporting events became evident, which was the main reason for creating the web application.

**Keywords:** Sports, Practice, Project, Organization, Matches.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Aplicativo Appito.....	22
Figura 2 – Aplicativo Fintta.....	23
Figura 3 – Aplicativo Fintta.....	23
Figura 4 – Aplicativo Chega+.....	24
Figura 5 – Aplicativo Chega+.....	24
Figura 6 – Gráfico de Respostas da Questão 01.....	34
Figura 7 – Gráfico de Respostas da Questão 02.....	35
Figura 8 – Gráfico de Respostas da Questão 03.....	35
Figura 9 – Gráfico de Respostas da Questão 04.....	36
Figura 10 – Gráfico de Respostas da Questão 05.....	36
Figura 11 – Scrum.....	38
Figura 12 – Estrutura Básica dos Sistemas.....	40
Figura 13 – Dimensões dos Sistemas de Informação.....	41
Figura 14 – Trello.....	42
Figura 15 – Arquitetura MVC.....	43
Figura 16 – Angular.....	47
Figura 17 – IntelliJ Idea.....	49
Figura 18 – Caso de Uso.....	61
Figura 19 – Protótipo Manter Jogador.....	63
Figura 20 – Protótipo Autenticar Jogador.....	65
Figura 21 – Protótipo Criar Partida.....	67
Figura 22 – Protótipo Buscar Partida.....	69
Figura 23 – Protótipo Inscrever-se na Partida.....	70
Figura 24 – Protótipo Visualizar Histórico de Partidas.....	72
Figura 25 – Protótipo Visualizar Partida.....	73
Figura 26 – Protótipo Gerenciar Partida.....	75
Figura 27 – Protótipo Sortear Times.....	77
Figura 28 – Protótipo Cronometrar Partida.....	78
Figura 29 – Protótipo Gerar Estatísticas da Partida.....	80
Figura 30 – Modelo Entidade Relacionamento.....	81
Figura 31 – Diagrama BPMN.....	82

Figura 32 – Diagrama BPMN.....	82
Figura 33 – Diagrama de Classe.....	83

## LISTA DE QUADROS

Quadro 1 – Sistemas Correlatos.....	25
Quadro 2 – Principais Envolvidos.....	27
Quadro 3 – Riscos.....	28
Quadro 4 – Cronograma.....	30
Quadro 5 – Tecnologias Back-end.....	45
Quadro 6 – Tecnologias Front-end.....	45
Quadro 7 – Outras Ferramentas.....	46
Quadro 8 – Endpoints API.....	55
Quadro 9 – Requisitos Funcionais.....	59
Quadro 10 – Requisitos Não Funcionais.....	60
Quadro 11 – Especificação de Caso de Uso Manter Jogador.....	62
Quadro 12 – Especificação de Caso de Uso Autenticar Jogador.....	64
Quadro 13 – Especificação de Caso de Uso Criar Partida.....	65
Quadro 14 – Especificação de Caso de Uso Buscar Partida.....	67
Quadro 15 – Especificação de Caso de Uso Inscrever-se na Partida.....	69
Quadro 16 – Especificação de Caso de Uso Visualizar Histórico de Partidas.....	71
Quadro 17 – Especificação de Caso de Uso Visualizar Partida.....	72
Quadro 18 – Especificação de Caso de Uso Gerenciar Partida.....	74
Quadro 19 – Especificação de Caso de Uso Sortear Times.....	75
Quadro 20 – Especificação de Caso de Uso Cronometrar Partida.....	77
Quadro 21 – Especificação de Caso de Uso Gerar Estatísticas da Partida.....	79

## LISTA DE ABREVIATURAS E SIGLAS

### ABREVIATURAS

Art.	Artigo
Obs.	Observação
s.d	Sem Data

### SIGLAS

API	Application Programming Interface
APP	Aplicativo
BPMN	Business Process Modeling Notation
DI	Dependency Injection
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IaaS	Infraestrutura como Serviço
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	Integrated Development Environment
IoC	Inversion of Control
JPQL	Java Persistence Query Language
Js	Java Script
JVM	Java Virtual Machine
MER	Modelo de entidade relacionamento
MVC	Model-View-Controller
PMBOK	Project Management Body of Knowledge
SaaS	Software como Serviço
SQL	Structured Query Language
TI	Tecnologia da Informação
UML	Unified Modeling Language
WEB	World Wide Web
XML	Extensible Markup Language

## SUMÁRIO

1.	INTRODUÇÃO .....	18
2.	CONTEXTUALIZAÇÃO .....	19
3.	PROPOSTA DE SOLUÇÃO .....	20
4.	OBJETIVOS .....	20
4.1.	Objetivo Geral .....	20
4.2.	Objetivos Específicos .....	21
5.	TRABALHOS CORRELATOS .....	21
6.	ESCOPO DO PROJETO .....	25
7.	NÃO ESCOPO .....	26
8.	PÚBLICO-ALVO .....	26
9.	PRINCIPAIS ENVOLVIDOS .....	27
10.	RISCOS .....	28
11.	PREMISSAS E RESTRIÇÕES .....	29
11.1.	Premissas .....	29
11.2.	Restrições .....	29
12.	CRONOGRAMA .....	29
13.	METODOLOGIA DE PESQUISA E TEMAS PRINCIPAIS .....	30
13.1.	Pesquisa de Campo .....	30
13.1.1	Tipo de Pesquisa .....	30
13.1.2	Coleta de Dados .....	31
13.1.3	Estrutura do Questionário .....	31
13.1.4	Amostragem .....	32
13.1.5	Análise dos Dados .....	32
13.1.6	Justificativa da Escolha da Ferramenta .....	33
13.1.7	Limitações da pesquisa .....	33
13.1.8	Resultados .....	34

<b>13.2.</b>	<b>Método de Gestão Ágil de Desenvolvimento – Scrum.....</b>	<b>37</b>
<b>13.2.1</b>	<b>Estrutura do Scrum .....</b>	<b>37</b>
<b>13.2.2</b>	<b>Artefatos do Scrum .....</b>	<b>38</b>
<b>13.3.</b>	<b>Engenharia de Software.....</b>	<b>38</b>
<b>13.4.</b>	<b>Sistema de Informação .....</b>	<b>39</b>
<b>13.5.</b>	<b>Trello.....</b>	<b>41</b>
<b>13.5.1</b>	<b>Características do Trello.....</b>	<b>42</b>
<b>13.5.2</b>	<b>Arquitetura do Trello .....</b>	<b>42</b>
<b>14.</b>	<b>CARACTERÍSTICAS TÉCNICAS DA SOLUÇÃO .....</b>	<b>43</b>
<b>14.1.</b>	<b>Arquitetura MVC .....</b>	<b>43</b>
<b>14.2.</b>	<b>Sistema Web .....</b>	<b>43</b>
<b>14.3.</b>	<b>Computação em Nuvem.....</b>	<b>44</b>
<b>14.4.</b>	<b>Quadro Resumo das Ferramentas Utilizadas .....</b>	<b>45</b>
<b>14.5.</b>	<b>Ferramentas e Tecnologias Utilizadas no Front-End .....</b>	<b>46</b>
<b>14.5.1</b>	<b>Angular.....</b>	<b>46</b>
<b>14.5.2</b>	<b>Node.Js .....</b>	<b>47</b>
<b>14.5.3</b>	<b>Visual Studio Code.....</b>	<b>48</b>
<b>14.6.</b>	<b>Ferramentas e Tecnologias Utilizadas no Back-End.....</b>	<b>48</b>
<b>14.6.1</b>	<b>IntelliJ Idea.....</b>	<b>48</b>
<b>14.6.2</b>	<b>Java .....</b>	<b>49</b>
<b>14.6.3</b>	<b>Spring Boot.....</b>	<b>50</b>
<b>14.6.4</b>	<b>Lombok .....</b>	<b>51</b>
<b>14.6.5</b>	<b>Maven .....</b>	<b>51</b>
<b>14.6.6</b>	<b>Conectividade com Banco de Dados – Hibernate .....</b>	<b>52</b>
<b>14.6.7</b>	<b>Segurança nas suas Aplicações - Spring Security .....</b>	<b>52</b>
<b>14.6.8</b>	<b>Postman .....</b>	<b>53</b>

14.6.9	Git .....	53
14.6.10	GitHub .....	54
14.6.11	API .....	54
14.6.12	MySQL .....	58
14.6.13	Draw.io.....	58
15.	ESPECIFICAÇÃO DE REQUISITOS .....	58
16.	CASO DE USO .....	60
16.1.	Diagrama de Caso de Uso .....	60
16.2.	Especificação dos Casos de Usos.....	61
16.2.1	Manter Jogador.....	62
16.2.1.1	<i>Detalhamento de Caso de Uso</i> .....	62
16.2.1.2	<i>Protótipo</i> .....	63
16.2.2	Autenticar Jogador.....	63
16.2.2.1	<i>Detalhamento de Caso de Uso</i> .....	64
16.2.2.2	<i>Protótipo</i> .....	65
16.2.3	Criar Partida .....	65
16.2.3.1	<i>Detalhamento de Caso de Uso</i> .....	65
16.2.3.2	<i>Protótipo</i> .....	66
16.2.4	Buscar Partida .....	67
16.2.4.1	<i>Detalhamento do Caso de Uso</i> .....	67
16.2.4.2	<i>Protótipo</i> .....	68
16.2.5	Inscriver-se na Partida .....	69
16.2.5.1	<i>Detalhamento do Caso de Uso</i> .....	69
16.2.5.2	<i>Protótipo</i> .....	70
16.2.6	Visualizar Histórico de Partidas .....	71
16.2.6.1	<i>Detalhamento do Caso de Uso</i> .....	71



16.2.6.2	<i>Protótipo</i>	72
16.2.7	Visualizar Partida	72
16.2.7.1	<i>Detalhamento do Caso de Uso</i>	72
16.2.7.2	<i>Protótipo</i>	73
16.2.8	Gerenciar Partida	74
16.2.8.1	<i>Detalhamento do Caso de Uso</i>	74
16.2.8.2	<i>Protótipo</i>	75
16.2.9	Sortear Times	75
16.2.9.1	<i>Detalhamento do Caso de Uso</i>	75
16.2.9.2	<i>Protótipo</i>	76
16.2.10	Cronometrar Partida	77
16.2.10.1	<i>Detalhamento do Caso de Uso</i>	77
16.2.10.2	<i>Protótipo</i>	78
16.2.11	Gerar Estatísticas da Partida	78
16.2.11.1	<i>Detalhamento do Caso de Uso</i>	79
16.2.11.2	<i>Protótipo</i>	79
17.	ARTEFATOS	80
17.1.	Modelo Entidade Relacionamento (MER)	80
17.2.	BPMN	81
17.3.	Diagrama de Classe	82
18.	CONSIDERAÇÕES FINAIS	84
19.	REFERÊNCIAS	86

## 1. INTRODUÇÃO

Atualmente, estabelecer uma rotina ativa e saudável é imprescindível para o combate ao sedentarismo, um problema crescente em diversas regiões. No Distrito Federal, os dados apontam um cenário desafiador em relação à prática insuficiente de esportes. Segundo o Ministério da Saúde (2023), o sistema de monitoramento Vigitel relata que 37% dos adultos da capital não atingem os 150 minutos semanais recomendados de atividade física moderada. Nesse contexto, como destacado por Samulski (2000), despertou-se na população a importância sobre a necessidade da inserção de novos hábitos, que contribuam para uma vida mais saudável e equilibrada, com o auxílio da prática esportiva como parte integrante do cotidiano.

A noção de qualidade de vida está relacionada aos ambientes em que as pessoas frequentam e à sua relação com a natureza. A partir disso, constata-se que a prática de esportes é fundamental para auxiliar na socialização, no trabalho em equipe e na melhoria da saúde física e mental.

Entretanto, em 2020, segundo o Instituto Brasileiro de Geografia e Estatística (IBGE, 2020), percebe-se que há dificuldades ao reunir jogadores suficientes para organizar partidas, uma vez que a falta de infraestrutura adequada e até mesmo de ferramentas que ajudem na organização das partidas, desanimam os jogadores a procurarem esses eventos. De frente com esses fatos, a tecnologia chega como um grande alicerce para o aprimoramento dessas problemáticas, trazendo assim soluções que conseguem otimizar boa parte desse processo de organização e gestão de eventos esportivos.

Nesse sentido, esta aplicação a ser apresentada tem o intuito de incentivar a prática esportiva, possibilitando o jogador organizar suas próprias partidas ou, se preferir, buscar jogos por meio do filtro de localização, facilitando o acesso a partidas próximas.

Assim, a implementação do sistema busca simplificar a organização de jogos, fornecendo ferramentas auxiliares tanto para a criação quanto para a gestão das partidas. Além disso, permite acompanhar as estatísticas pós-jogos, promovendo uma experiência mais completa.

## 2. CONTEXTUALIZAÇÃO

De acordo com dados do IBGE (2020), cerca de 61,3 milhões de brasileiros praticam algum tipo de esporte ou atividade física no Brasil, refletindo a importância da prática esportiva no cotidiano de cada integrante. Dentre as modalidades mais populares, o futebol é a atividade com o maior destaque no país, seguido por caminhadas e atividades fitness. Esses números mostram não apenas o aspecto recreativo, mas também a busca por melhorias na saúde e no bem-estar, já que 40,2% dos praticantes afirmam que essas atividades contribuem diretamente para a qualidade de vida. Com relação à frequência, o estudo revelou que 26,3% dos participantes realizam atividades esportivas pelo menos quatro vezes por semana, entretanto, outros 7,8% relatam praticar pelo menos uma vez por semana, o que sugere a necessidade de iniciativas para aumentar o incentivo de uma prática mais regular dessas atividades esportivas entre os brasileiros.

Com o aumento significativo na demanda por partidas esportivas acessíveis e organizadas, muitos jogadores encontram desafios para encontrar grupos ou jogos disponíveis. Uma das principais barreiras é localizar partidas, já que a divulgação é realizada através de convites pessoais ou redes sociais, o que limita o alcance a um círculo de amigos e conhecidos, dificultando a possibilidade de conexão com novos jogadores. Outro obstáculo comum é que, por muitas vezes, não têm um contingente necessário de pessoas, o que inviabiliza a realização das partidas. Por fim, a localização acaba sendo um empecilho, pois nem sempre é fácil encontrar partidas perto à residência ou localização atual dos jogadores.

Enquanto, para aqueles que organizam as partidas, a gestão costuma ser manual, através de folhas de papel ou grupos em redes sociais. Esse processo pode ser desorganizado e demorado, impactando na eficiência das partidas.

Diante desse cenário, o desenvolvimento de um sistema para a organização de partidas esportivas surge como uma solução necessária. A plataforma busca melhorar a experiência dos usuários com funcionalidades, como filtro de localização, preferências esportivas, dentre outras. Além de garantir uma gestão mais eficiente, incentiva a prática de esportes mais acessível e regular, promovendo a integração social. Dessa forma, o sistema auxilia no fortalecimento da comunidade esportiva, contribuindo com a saúde e promovendo o bem-estar coletivo.

### **3. PROPOSTA DE SOLUÇÃO**

O Resenha Esporte Clube surgiu com o objetivo de apoiar jogadores a encontrarem e a exporem suas partidas de forma mais fácil e centralizada, trazendo funcionalidades para melhorar a experiência dos usuários nas suas partidas.

Para ter acesso à plataforma, o usuário tem que se cadastrar no sistema gratuitamente. Dessa forma será criado o perfil de “Jogador” e as seguintes funcionalidades serão disponibilizadas:

- Criar partidas: Possibilita criar listas para que os jogadores se inscrevam para participar de partidas do esporte ofertado;
- Buscar partidas: Possibilita encontrar partidas no Distrito Federal por meio de filtros;
- Realizar inscrição: Permite que o usuário encontre uma partida e se inscreva para participar sem grandes burocracias;
- Sorteio: Após a lista estar completa, será possível sortear os times aleatoriamente;
- Cronômetro: De acordo com o esporte praticado, o sistema pode disponibilizar um cronômetro para a marcação do tempo;
- Estatísticas: Ao término da partida, as estatísticas do jogo serão geradas com base nas pontuações coletadas por meio das vitórias, empates e derrotas.

Quanto à garantia de um sistema mais acessível, optou-se pelo desenvolvimento de uma aplicação web, que pode ser acessada por diversos dispositivos como celulares, tablets e desktops. O objetivo é alcançar um público mais amplo e sem restrições quanto ao tipo de dispositivo utilizado.

### **4. OBJETIVOS**

#### **4.1. Objetivo Geral**

Desenvolver uma solução web que visa apoiar a organização de práticas esportivas de basquete, futebol, futevôlei e vôlei. Permitindo encontrar jogadores para

a composição dos times, além de oferecer um controle maior das partidas e dos jogadores participantes.

#### 4.2. Objetivos Específicos

Levando em conta as etapas para o alcance do objetivo geral, os seguintes pontos constituem os objetivos específicos:

- Mapear e identificar as reais necessidades dos jogadores em relação à organização de partidas, bem como realizar a análise de requisitos;
- Desenvolver e validar o protótipo com os potenciais usuários do sistema;
- Codificar o sistema conforme os requisitos e os protótipos validados;
- Testar todas as funcionalidades do sistema para garantir a qualidade;
- Apresentar o sistema funcional aos potenciais usuários.

### 5. TRABALHOS CORRELATOS

Neste tópico apresenta-se uma análise de sistemas correlatos nas suas características e funcionalidades, inclusive o do sistema “Resenha Esporte Clube”.

<sup>1</sup>**Appito:** O Appito é um aplicativo mobile que tem como missão conectar pessoas por meio do esporte, unindo a paixão pelo futebol à tecnologia e ao entretenimento, com o objetivo de proporcionar experiências únicas para os usuários. Além do aplicativo, a empresa oferece serviços de aluguel de arenas esportivas e escolas de futebol (APITTO, s.d), permitindo que jogadores encontrem partidas públicas e privadas para participarem. O sistema também permite o registro de estatísticas pessoais e a gestão de jogos, oferecendo funcionalidades como a criação de partidas, envio de convites e busca por jogadores, entre outras funcionalidades.

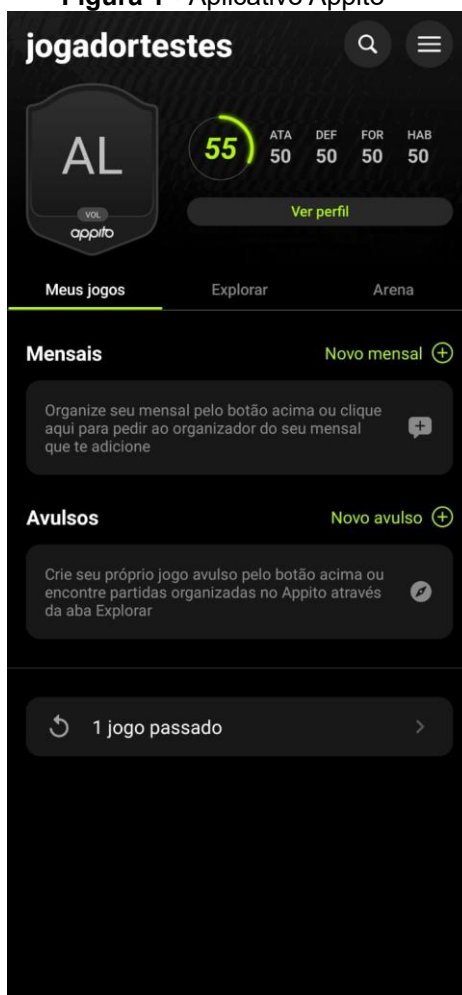
Por mais que a plataforma facilite a busca de pessoas para completar as partidas esportivas, o aplicativo limita-se, exclusivamente, ao futebol, trazendo uma plataforma completa apenas para um nicho específico, deixando outros esportes, como o vôlei, sem o mesmo apoio. Além disso, por ser um aplicativo mobile, restringe

---

<sup>1</sup> Appito. Appito, Data não Especificada. É o aplicativo perfeito para que você possa organizar seu futebol, do começo ao fim. Disponível em: [PlayStore:https://play.google.com/store/apps/details?id=com.apitador.app](https://play.google.com/store/apps/details?id=com.apitador.app). Acesso em: 20 set. 2024.

os usuários a acessarem o sistema por outros aparelhos eletrônicos que não sejam celulares ou tablets, diferentemente de uma aplicação web que pode ser acessada por qualquer dispositivo que tenha um navegador.

**Figura 1 - Aplicativo Appito**



**Fonte: Autores, 2024**

<sup>2</sup>**Fintta:** O Fintta é um aplicativo mobile que, segundo a desenvolvedora Fintta (2024), oferece uma gestão completa de partidas de futebol, combinando facilidade para os organizadores com funcionalidades abrangentes para os jogadores, em um único aplicativo. Ele permite a organização rápida das partidas, facilita a busca por

---

<sup>2</sup> Fintta. Fintta, Data não Especificada. Convoque os amigos para disputar as melhores partidas. Organize equipes, agende jogos, grave os lances com o fintta. Disponível em: PlayStore: <https://play.google.com/store/apps/details?id=com.fintta.app> App Store: <https://apps.apple.com/us/app/fintta-play-soccer-today/id934328306>. Acesso em: 20 set. 2024.

jogos, gera estatísticas detalhadas e fornece uma maneira inovadora de avaliação dos jogadores.

No entanto, assim como o Appito, exemplo citado anteriormente, o Fintta atende, exclusivamente, ao nicho do futebol, limitando aos esportes o uso da ferramenta para organizar suas partidas. Além disso, por ser um aplicativo mobile, dificulta a acessibilidade e o alcance da plataforma, restringindo o seu uso àqueles que tem instalado o aplicativo em seus dispositivos.

**Figura 2 e 3 - Aplicativo Fintta**



Fonte: Autores, 2024

<sup>3</sup>**Chega+:** O Chega+ é um aplicativo para a organização de grupos de futebol, abrangendo desde o sorteio até a definição da tabela de jogos. O aplicativo, desenvolvido por brasileiros e lançado no Brasil, possibilita que todos os jogadores de um grupo participem do gerenciamento de partidas, confirmem a presença,

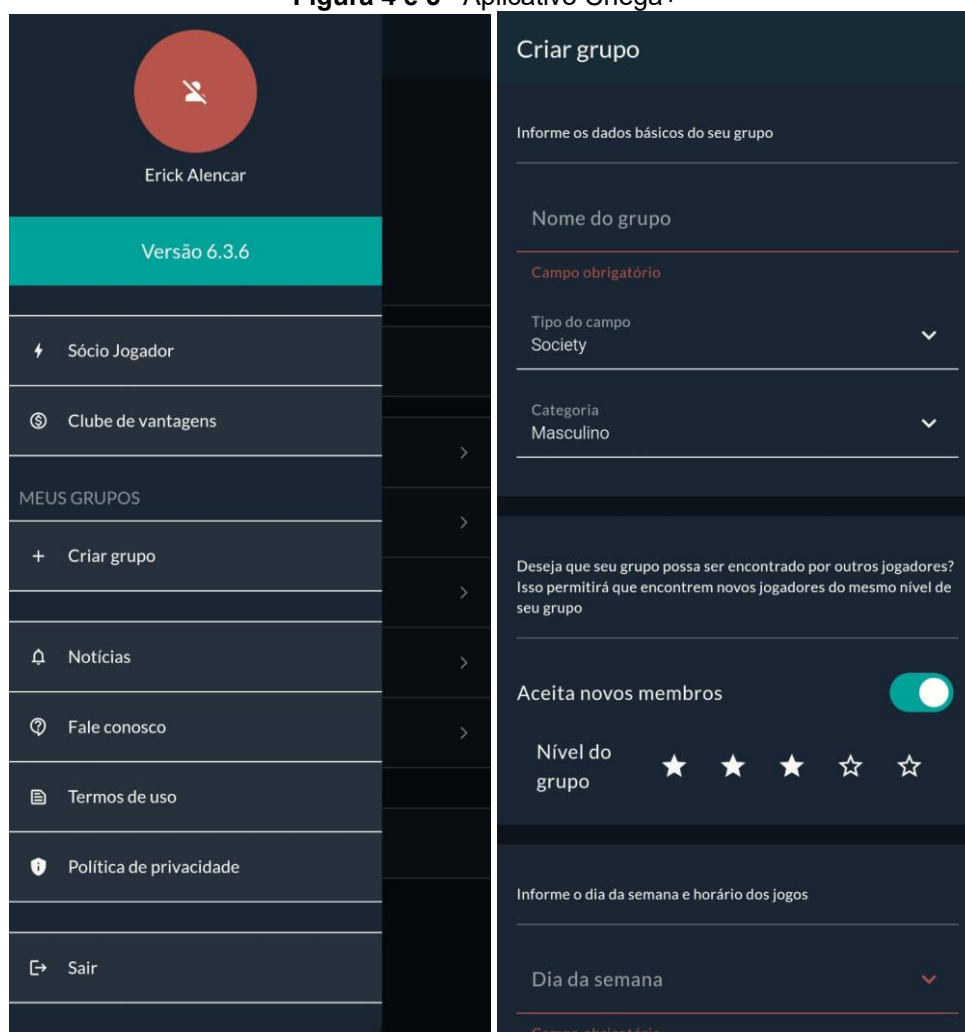
---

Chega+. Chega+, Data não Especificada. O Chega+ é o melhor aplicativo do Mundo para organizar grupos de futebol. PlayStore: <https://play.google.com/store/apps/details?id=app.chegamais.com.chegamais>. Acesso em: 20 set. 2024.

acompanhem o placar, além de diversas outras funcionalidades que facilitam a gestão dos jogos.

Todavia, o Chega+ é um aplicativo voltado, exclusivamente, ao futebol, podendo apenas mudar a modalidade em que o esporte é jogado. Embora ofereça diversas funcionalidades, a maioria delas são pagas, sendo necessário a realização de assinaturas mensais ou anuais para ter acesso a elas. Dessa forma, a plataforma oferece poucas funcionalidades aos usuários de assinatura gratuita e, por ser um aplicativo mobile, o seu uso é restrito àqueles que o tenham instalado em seus dispositivos.

**Figura 4 e 5 - Aplicativo Chega+**



Fonte: Autores, 2024



**Quadro 1 - Sistemas Correlatos**

<b>Funcionalidades</b>	<b>Appito</b>	<b>Fintta</b>	<b>Chega+</b>	<b>Resenha Esporte Clube</b>
Criar partidas de esportes variados	Não	Não	Não	Sim
Buscar partidas	Sim	Sim	Sim	Sim
Inscrever-se em partidas	Sim	Sim	Sim	Sim
Sorteio de times	Sim	Sim	Sim	Sim
Cronômetro	Sim	Sim	Sim	Sim
Estatísticas da partida	Sim	Sim	Sim	Sim
Visualizar histórico de partidas	Sim	Sim	Sim	Sim
Multiplataforma	Não	Não	Não	Sim

Fonte: Autores, 2024

## 6. ESCOPO DO PROJETO

Para este projeto fica definido que, a estrutura do escopo inicial será disponibilizada, conforme definido nos itens abaixo:

- Será disponibilizada uma aplicação web responsiva, trazendo assim a possibilidade de acesso por dispositivos móveis;
- Otimização da interface para garantir uma boa experiência em diferentes resoluções de telas;
- É disponibilizado pelo sistema uma autenticação para os usuários, possibilitando a criação de contas e a realização de logins para o acesso às funcionalidades. Poderão se registrar no sistema através de um formulário com informações básicas como: nome, apelido, data de nascimento, e-mail, telefone e senha;
- Os usuários poderão registrar novas partidas preenchendo um formulário com informações, como: nome da partida, tipo de modalidade, número máximo de jogadores, data, local, horário, descrição, endereço e região administrativa;

- Os usuários terão a possibilidade de visualizar e buscar partidas disponíveis, verificar a data, o local e o horário, além de visualizar a quantidade de vagas disponíveis em cada partida;
- Os usuários, após localizar partidas em aberto, poderão se inscrever naquelas que desejarem;
- O sistema, ao disponibilizar um perfil de jogador, permite ao usuário gerenciar suas partidas, incluindo a edição de informações dos jogos, a formação de times através de sorteio, o uso do cronômetro e a marcação de placares;
- É disponibilizado pelo sistema, o acesso do usuário ao histórico de partidas jogadas e às estatísticas dos jogos;
- O sistema atende à Lei Geral de Proteção de Dados (LGPD).

## 7. NÃO ESCOPO

- Nenhum requisito adicional que não esteja projetado nesse trabalho será implementado. Não serão adicionadas funcionalidades que não esteja relacionada a pesquisa de partidas, organização de partidas e consultar estatísticas;
- Não fazem parte do escopo: o desenvolvimento de um *app mobile*, recursos para a criação de partidas públicas e privadas, possibilidade de reservar quadras, fornecimento de anúncios, inserção de outras modalidades esportivas diferentes da do sistema, envio de e-mails e implementação de uma API de geolocalização.

## 8. PÚBLICO-ALVO

O projeto engloba um grupo principal de consumidores que irá utilizar o sistema tanto para buscar partidas quanto para organizá-las.

Esse grupo é composto por cidadãos e jogadores, assegurando a esse público um acesso rápido e organizado às partidas, permitindo a prática de esportes de forma planejada e sem complicações.

## 9. PRINCIPAIS ENVOLVIDOS

**Quadro 2 - Principais Envolvidos**

<b>Envolvidos</b>	<b>Papel</b>	<b>Interesse</b>
Público-alvo I	Trata-se daquele jogador que usará o sistema para buscar partidas esportivas para lazer.	O interesse deste público é encontrar partidas através de filtros de modalidade esportiva e localidade desejados para a prática de esportes.
Público-alvo II	Tem como foco jogadores que queiram criar e gerenciar suas partidas esportivas.	O interesse deste é conseguir criar partidas esportivas, divulgar a mesma para conseguir jogadores suficientes e gerenciá-la.
Erick Alencar Larissa Novais Rafael Caetano	Front-End	Implementação das telas e identidade visual do sistema.
Erick Alencar Larissa Novais Rafael Caetano	Back-End	Implementação das regras de negócio do sistema.
Erick Alencar Larissa Novais Rafael Caetano	Banco de Dados	Modelagem de banco de dados do sistema.
Erick Alencar Larissa Novais Rafael Caetano	Documentação	Documentar todas as etapas do desenvolvimento do sistema.
Me. Gleidson Porto	Orientador	Orientação, avaliação e condução de ideias propostas no trabalho e no sistema desenvolvido.

**Fonte: Autores, 2024**

## 10. RISCOS

Os riscos são ocorrências incertas que podem provocar um efeito positivo ou negativo a um projeto. De acordo com o Ferreira (2013), não existe um projeto se quer que não sofra com alguma incerteza no decorrer do tempo.

A tabela abaixo apresenta os riscos identificados no sistema Resenha Esporte Clube, assim como os responsáveis pelo tratamento do risco, bem como as ações propostas para resolvê-los.

**Quadro 3 - Riscos**

Risco	Probabilidade	Impacto	Responsável	Ações	Período
Não aceitabilidade e do sistema pelo público-alvo I e II.	Baixa	Alto	Gerente do Projeto	Acompanhar o cadastro de usuários no sistema. Caso a adesão seja baixa, realizar campanhas de publicidade para melhorar a sua divulgação.	Diário
Não comparecimento de jogadores nas partidas.	Médio	Médio	Público-alvo I	Comunicar imediatamente e explicar o real motivo do não comparecimento ao público-alvo II.	Diário
Cancelamento das partidas em cima da hora.	Médio	Médio	Público-alvo II	Comunicar imediatamente e explicar o real motivo do cancelamento ao público-alvo I.	Diário
Sistema estar fora do ar.	Baixo	Alto	Gerente do Projeto	Implementar planos de contingência, caso tenha algum problema. Após a correção, monitorar e testar para garantir a funcionalidade correta do sistema.	Diário

Quebra do sigilo dos usuários.	Médio	Alto	Gerente do Projeto	Contratação de uma empresa para verificar a segurança dos dados.	Mensal
--------------------------------	-------	------	--------------------	--	--------

Fonte: Autores, 2024

## 11. PREMISSAS E RESTRIÇÕES

### 11.1. Premissas

Segundo o Guia PMBOK (*Project Management Body of Knowledge*), as premissas de um projeto são fatores hipotéticos, utilizados na fase do planejamento, que podem ser considerados verdadeiros, reais ou certos, sem a necessidade da realização de prova e demonstração. Diante disso, definimos como premissas os seguintes itens:

- A aceitabilidade do sistema pelo público-alvo I e II (usuários do sistema);
- O interesse em obter as ferramentas oferecidas para facilitar a criação e gerenciamento de partidas (público-alvo II);
- A necessidade de ferramentas para o gerenciamento de partidas na prática esportiva.

### 11.2. Restrições

De acordo com o Guia PMBOK (*Project Management Body of Knowledge*), as restrições de um projeto são fatores limitadores que impactam na sua execução e no seu processo. Diante disso, definimos como restrições os seguintes itens:

- **Prazo:** por existir um tempo de desenvolvimento do projeto curto, os prazos propostos devem ser respeitados;
- **Custos:** por ser um sistema para fins acadêmicos, não houve custos e todas as tecnologias utilizadas foram gratuitas.

## 12. CRONOGRAMA

O cronograma de entrega e execução das atividades foi elaborado para assegurar uma gestão eficiente no desenvolvimento e conclusão do projeto. As tarefas foram divididas em etapas, contendo data de entrega e duração bem definidas. Esse detalhamento não visa apenas atingir prazos, mas também uma entrega final de alta qualidade dentro do período estipulado.

**Quadro 4 - Cronograma**

<b>CRONOGRAMA DE ENTREGA/EXECUÇÃO (etapas e data de entrega)</b>				
<b>Nome</b>	<b>Atividade</b>	<b>Data Início</b>	<b>Data Fim</b>	<b>Duração</b>
Erick Alencar, Larissa Novais e Rafael Caetano	Planejamento do Projeto.	12/08/2024	16/08/2024	5 dias
Erick Alencar, Larissa Novais e Rafael Caetano	Levantamento de Requisitos.	17/08/2024	20/08/2024	4 dias
Erick Alencar, Larissa Novais e Rafael Caetano	Diagrama de caso de uso, diagrama de classes e diagrama bpmn.	08/10/2024	14/10/2024	7 dias
Erick Alencar, Larissa Novais e Rafael Caetano	Modelar banco de dados.	08/10/2024	10/10/2024	3 dias
Erick Alencar, Larissa Novais e Rafael Caetano	Implementar o Back-End.	11/10/2024	13/11/2024	32 dias
Erick Alencar, Larissa Novais e Rafael Caetano	Implementar o Front-End.	11/10/2024	13/11/2024	32 dias
Erick Alencar, Larissa Novais e Rafael Caetano	Teste do sistema e preparação para defesa do projeto.	13/11/2024	30/11/2024	17 dias
Erick Alencar, Larissa Novais e Rafael Caetano	Elaboração do projeto de conclusão de curso.	02/09/2024	03/11/2024	63 dias

**Fonte: Autores, 2024**

## **13. METODOLOGIA DE PESQUISA E TEMAS PRINCIPAIS**

### **13.1. Pesquisa de Campo**

Para alcançarmos melhores resultados na coleta de dados importantes, que mostram o melhor caminho para o desenvolvimento da solução web, foi utilizada a metodologia de pesquisa. Essa aplicação tem o objetivo de identificar as principais necessidades enfrentadas por jogadores e organizadores de partidas esportivas, além de coletar sugestões de funcionalidades que os usuários consideraram importantes para a implementação na plataforma através da pesquisa.

#### **13.1.1 Tipo de Pesquisa**

O tipo de pesquisa adotado nesse levantamento foi a qualitativa descritiva, que tem a intenção de trazer opiniões, dificuldades e necessidades do

público-alvo. Conforme citado por Gil (2007), a pesquisa descritiva tem como alvo “descrever as características de determinada população ou fenômeno, sem, contudo, manipular variáveis”. Dito isso, tem como objetivo filtrar um perfil de público interessado na solução, assim também, detectar as funcionalidades mais requisitadas para melhorar a experiência do usuário em organizar e participar de partidas esportivas.

### **13.1.2 Coleta de Dados**

Para essa captação de dados, utilizou-se um questionário *online* através da ferramenta do *Google Forms*. A escolha foi feita por ser uma plataforma gratuita e de fácil acesso e manuseio, além de possibilitar a criação de questionários personalizados com diversas perguntas, como de múltipla escolha, escala de ação, respostas abertas, entre outras.

O questionário foi separado por duas partes principais:

1. Dados Demográficos e Perfil dos Participantes: Foram coletados dados dos participantes como opiniões, funcionalidades, esportes praticados e frequência que vão às partidas esportivas. Esses dados são importantes para entender o perfil do público-alvo da plataforma e assim realizar adaptações e trazer soluções de acordo com suas necessidades;
2. Experiência e Dificuldades na Organização de Partidas: Foram realizadas perguntas sobre as principais dificuldades que os participantes acabam encontrando ao participar ou organizar uma partida. Além de também ser questionadas questões sobre funcionalidades consideradas importantes para a plataforma, como controle de inscrições, estatísticas gerais entre outros.

### **13.1.3 Estrutura do Questionário**

O questionário foi estruturado de forma simples e direta, permitindo que os participantes o respondessem em, aproximadamente, 5 a 10 minutos. Segue abaixo um breve resumo das perguntas realizadas:

1. Perfil do Participante:

- Qual desses esportes você pratica?
2. Dificuldades na Organização de Partidas:
- Possui alguma dificuldade para encontrar partidas para jogar no seu momento de lazer?
  - Como você encontra partidas para jogar?
  - Em sua opinião, um sistema para buscar e organizar partidas poderia facilitar a prática de esportes?
3. Preferências de Funcionalidades:
- As funcionalidades do sistema, como sortear times, gerar estatísticas do jogo e cronômetro, seriam úteis para a criação de partidas?
  - Em sua opinião, há alguma funcionalidade que não pode faltar no sistema?

#### **13.1.4 Amostragem**

A pesquisa, que tem o objetivo de captar o maior número de opiniões e necessidades do público-alvo, foi composta por amostras de jogadores e organizadores de partidas. O formulário, que utilizou o método de amostragem não probabilística por conveniência, foi divulgado através de redes sociais, grupos de esportes e entre contatos pessoais que participam da atmosfera esportiva.

A amostra esperada foi de aproximadamente 15 a 30 participantes, número que traz uma base representativa para analisar as expectativas e as necessidades dos usuários em relação à proposta de solução.

#### **13.1.5 Análise dos Dados**

Serão realizadas análises quali-quantitativa, nas quais as respostas fechadas, de múltipla escolha ou de opinião, serão filtradas através de estatísticas descritivas, como frequências e porcentagens, para identificar as mais recorrentes entre os participantes. Enquanto as respostas discursivas, passarão por uma análise qualitativa, identificando padrões e divergências nas opiniões, filtrando ideias



relevantes para a construção da plataforma a fim de melhorar a experiência do usuário.

#### **13.1.6 Justificativa da Escolha da Ferramenta**

O *Google Forms* foi escolhido como a plataforma de coleta de dados devido a sua praticidade, tanto para a criação do formulário quanto para o fácil acesso do usuário. Algumas de suas vantagens são:

- **Gratuidade:** É uma ferramenta que elimina custos adicionais no processo da realização da pesquisa;
- **Facilidade de Manuseio:** A elaboração e a entrega de questionários são rápidas, sendo acessíveis em qualquer dispositivo com acesso à internet;
- **Relatórios Automatizados:** Gera relatórios automáticos com gráficos e tabelas, facilitando a visualização rápida dos dados coletados;
- **Armazenamento de Dados em Nuvem:** Todas as respostas são armazenadas automaticamente no Google Drive, garantindo a segurança e o fácil acesso às informações para a análise.

#### **13.1.7 Limitações da pesquisa**

Como qualquer outra pesquisa, esse estudo há algumas limitações, estas são:

- **Amostragem não probabilística:** O público-alvo foi escolhido de maneira não probabilística, o que indica que os resultados não podem ser generalizados para toda a população de praticantes de esportes. Todavia, a amostragem foi suficiente para fornecer dados valiosos sobre a necessidade dos usuários;
- **Respostas subjetivas:** Por mais que as perguntas tenham sido construídas para se chegar às respostas objetivas, há possibilidade de conter opiniões subjetivas que não representem o público-alvo.

Apesar dessas limitações, a metodologia utilizada é suficiente para a coleta de dados importantes sobre os perfis dos usuários e suas expectativas em relação à solução web, contribuindo para o desenvolvimento de um sistema que atenda às necessidades do público-alvo.

### 13.1.8 Resultados

Os resultados da pesquisa foram fundamentais para o direcionamento das funcionalidades da aplicação. Além disso, foi possível captar informações relevantes do que os usuários precisam em sua organização ou busca por práticas esportivas.

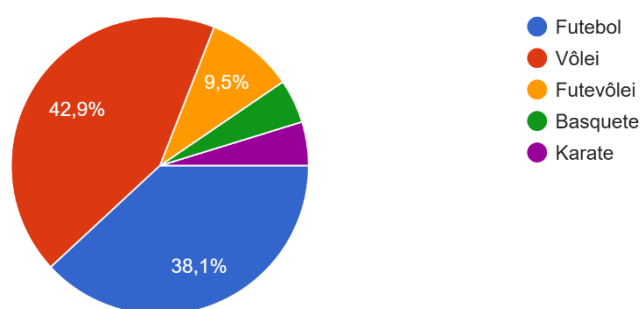
Durante a construção da questão abaixo, houve a intenção de obter quais esportes são mais praticados dentro do nosso público-alvo.

Foi apurado que o vôlei vem a frente com 42,9%, em seguida, respectivamente, vem o futebol com 38,1%, futevôlei com 9,6% e o basquete igualmente com karatê obteve 4,8% dos votos.

**Figura 6** - Gráfico de Respostas da Questão 01.

Qual desses esportes você pratica?

21 respostas



**Fonte: Autores, 2024**

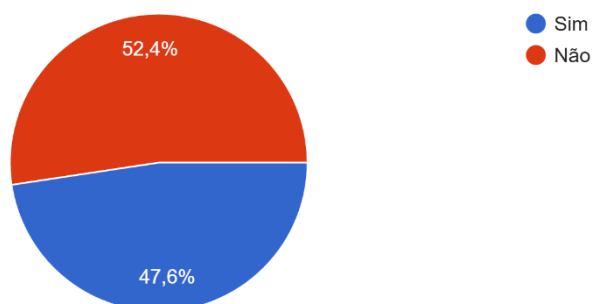
Na elaboração da questão abaixo, houve a intenção de identificar, ainda na visão do cidadão, se há dificuldades para a busca por partidas esportivas.

Obtém-se que 52,4% não sofrem com dificuldades, já 47,6% sofrem com a dificuldade de encontrar partidas.

**Figura 7** - Gráfico de Respostas da Questão 02.

Possui alguma dificuldade para encontrar partidas para jogar no seu momento de lazer?

21 respostas



**Fonte: Autores, 2024**

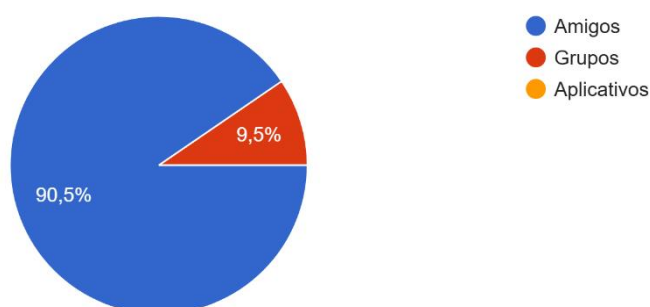
Na construção da questão abaixo, tem como objetivo identificar os principais meios que o público-alvo busca e organiza suas partidas esportivas.

Assim foi identificado que 90,5% utilizam de amigos e outros 9,5% usam de grupos de redes sociais, enquanto percebe-se que não há busca de partidas através de aplicativos.

**Figura 8** - Gráfico de Respostas da Questão 03.

Como você encontra partidas para jogar?

21 respostas



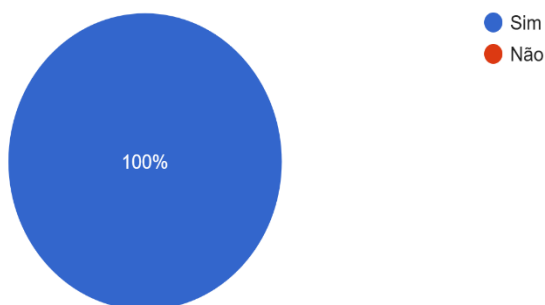
**Fonte: Autores, 2024**

Na elaboração da questão abaixo, com 100% das respostas, entende-se que há uma grande necessidade de uma aplicação que supra de forma eficaz à experiência de facilitar e melhorar as práticas esportivas.

**Figura 9-** Gráfico de Respostas da Questão 04.

Em sua opinião, um sistema para buscar e organizar partidas poderia facilitar a prática de esportes?

21 respostas



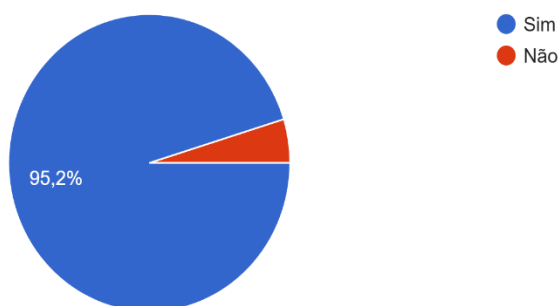
**Fonte: Autores, 2024**

Através de uma questão, que obteve 95,2% dos votos, conclui-se que as funcionalidades citadas são de grande importância na construção da aplicação e para suprir as necessidades do público-alvo.

**Figura 10-** Gráfico de Respostas da Questão 05.

As funcionalidades do sistema, como sortear times, gerar estatísticas do jogo e cronômetro, seriam úteis para a criação de partidas?

21 respostas



**Fonte: Autores, 2024**

E por fim, as respostas captadas na questão discursiva sobre as funcionalidades que o público-alvo consideram essenciais foram analisadas de forma

criteriosa, para assim identificar várias sugestões relevantes. Essas respostas foram organizadas e priorizadas, resultando em pautas específicas que ajudaram as decisões para a implementação de funcionalidades na aplicação. Esse processo assegurou que o desenvolvimento fosse alinhado às expectativas e necessidades reais dos usuários, contribuindo para a criação de uma aplicação mais funcional e atrativa.

### **13.2. Método de Gestão Ágil de Desenvolvimento – Scrum**

Atualmente, devido aos contínuos desafios organizacionais, surgiu-se a necessidade das organizações se manterem ativas e terem vantagens competitivas em relação aos seus concorrentes (MARZALL, 2016). Com isso, as empresas têm alterado a sua forma de gestão de demandas dentro de projetos que fogem do método tradicional, dando importância à adoção e à compreensão da metodologia ágil (STOPA, 2019).

Segundo Fábio Cruz (2013), o Scrum, framework iterativo e incremental, surgiu para auxiliar no gerenciamento de projetos de forma eficaz, focando na entrega de valor no menor tempo possível. Esse framework ágil contempla com autores e eventos fundamentais para o alcance do objetivo final dentro de cada *Sprint*, que é um período fixo de 2 a 4 semanas. Devido à sua flexibilidade e adaptabilidade durante o desenvolvimento da aplicação, o Scrum é amplamente utilizado na organização dos trabalhos em equipe, uma vez que o engajamento dos membros facilita a tomada de decisões e contribui para um desenvolvimento mais colaborativo.

Para garantir o bom funcionamento da equipe, foram atribuídos papéis a cada integrante. O Erick Alencar, atuando como Product Owner, foi responsável por gerenciar o projeto e definir as prioridades para a execução das atividades. A Larissa Novais, como Scrum Master, garantiu que os princípios do Scrum fossem seguidos, além de facilitar a comunicação entre os membros da equipe. E, por fim, o Rafael Caetano, como desenvolvedor, foi responsabilizado por se auto-organizar para tomar decisões e entregar os incrementos do produto. Com essa divisão de responsabilidades, percebeu-se durante o projeto uma execução eficiente e organizada, assegurando os princípios do Scrum.

#### **13.2.1 Estrutura do Scrum**

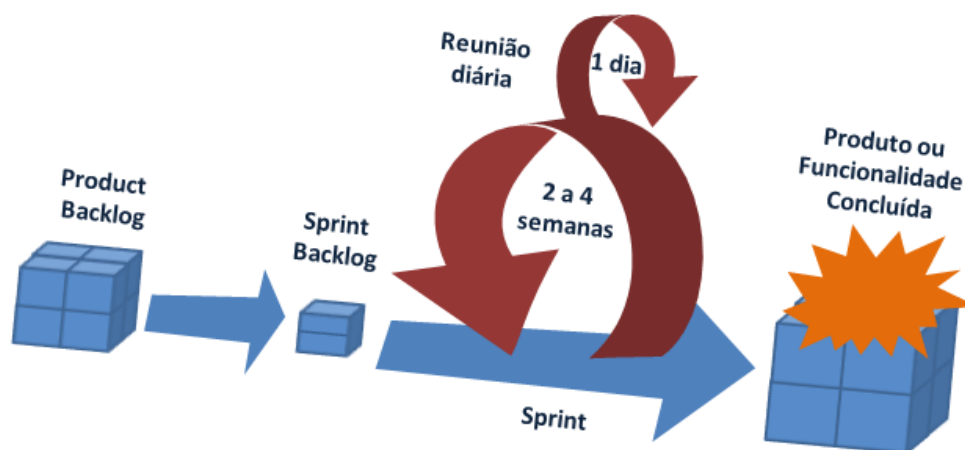
O Scrum pode ser adaptado às necessidades de cada organização durante a execução do projeto, diferenciando-se dos métodos tradicionais. A estrutura desse

método ágil é baseada no empirismo, onde afirma que o conhecimento vem das experiências e da observação. Partindo desse pressuposto, essa estrutura é utilizada para ajudar as equipes imaturas, que não sabem tudo no início do projeto, a se adaptarem e aprenderem, buscando melhorias contínuas através de suas experiências e nas tomadas de decisão com o auxílio da observação (ATLASSIAN, 2021).

### 13.2.2 Artefatos do Scrum

Os artefatos são as ferramentas que produzimos para solucionar os problemas. No Scrum existem três artefatos principais que estruturam o desenvolvimento do projeto. O primeiro é o *Product Backlog*, lista principal de atividades que devem ser feitas pela equipe do projeto e é administrado pelo *Product Owner* e *Scrum Master*, que vão elencar o grau de prioridade de cada atividade. Em seguida, há o *Sprint Backlog*, que consiste em uma seleção de itens do *Product Backlog* que devem ser implementadas durante o *Sprint*. Por fim, o Incremento é o produto final resultante do Sprint, que é entregue ao cliente (ATLASSIAN, 2021).

Figura 11 - Scrum



Fonte: Mindmasterbrasil, 2019

### 13.3. Engenharia de Software

A engenharia de software, definida também como uma disciplina, utiliza um conjunto de métodos, ferramentas e técnicas para projetar, analisar e gerenciar a

manutenção do desenvolvimento do software. Seu objetivo é produzir e manter software dentro dos prazos de metas, orçamentos e padrões de qualidade estabelecidos (CORTÉS, 2013).

Com a chegada da Crise do Software, o avanço da tecnologia diminuiu a necessidade de grandes investimentos para adquirir computadores domésticos. Esse fenômeno resultou em um aumento da demanda por novos softwares em um mercado despreparado para atender essa nova realidade. Nesse contexto, a intervenção da engenharia de software tornou-se necessária para solucionar essas problemáticas e atender às novas necessidades do setor (CORTÉS, 2013).

Esse projeto utilizou diversos princípios essenciais da engenharia de software para assegurar uma melhor qualidade e escalabilidade da solução. Esses princípios, baseados nos conceitos de Sommerville (2019), estruturam-se o seguinte ciclo de desenvolvimento de software:

- Requisitos: Fase na qual são feitos os alinhamentos com o cliente para as funcionalidades específicas do sistema;
- Desenvolvimento: Produzir em código de programação os planejamentos traçados dos requisitos definidos;
- Testes: Fase de validação para verificar se o sistema produzido atende aos requisitos;
- Implantação: Disponibilização do sistema aos usuários;
- Manutenção: Implementação e correção de funcionalidades do sistema de acordo com as novas necessidades.

Para o desenvolvimento do sistema Resenha Esporte Clube, foi considerado o modelo cascata, iniciando o projeto com levantamento de requisitos, seguidos por planejamento, modelagem, construção, entrega e suporte contínuo para o sistema.

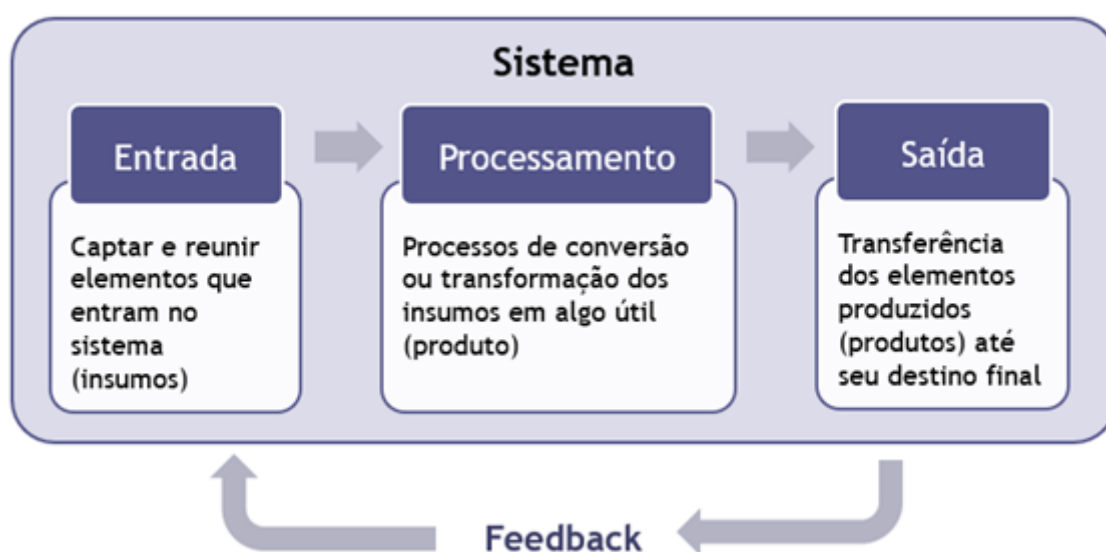
#### **13.4. Sistema de Informação**

Segundo Oliveira (1993), um sistema é definido como um conjunto de partes interativas e interdependentes que, em conjunto, formam uma unidade com um objetivo específico e efetuam funções determinadas. Por essa definição, é difícil conceber qualquer sistema que não produza algum tipo de informação, independentemente do nível e do tipo de uso.

Um sistema de informação é um conjunto bem estruturado de dados, pessoas, software, hardware e rede de comunicação, os quais são capturados e moldados em informações ao ambiente organizacional (O'BRIEN, 2004). Essa estrutura é uma ferramenta que converte dados em informações úteis, seja através de tecnologias ou processos manuais de entrada de dados.

De acordo com O'brien (2004), “um sistema pode ser definido simplesmente como um grupo de elementos inter-relacionados ou em interação que formam um todo unificado. Um sistema dessa ordem (às vezes chamado sistema dinâmico) possui três componentes ou funções básicas em interação”. Segue abaixo um exemplo da estrutura básica dos sistemas de informação, composto pelos elementos essenciais para o seu funcionamento.

**Figura 12 - Estrutura básica dos sistemas**



**Fonte: Bruna Diirr, 2021**

A entrada é responsável pela coleta e junção dos dados que entram no sistema. O processamento transforma esses dados que foram captados, convertendo-os em informações úteis, enquanto a saída realiza a transferência das informações e dos processamentos aos destinatários finais (O'BRIEN, 2004). Para garantir melhorias contínuas, o feedback é uma etapa crucial, que retorna as informações aos membros



da empresa para que seja avaliado e corrigido, garantindo o aprimoramento do sistema (LAUDON; LAUDON, 2009).

Segundo Laudon e Laudon (2009), o sistema de informação é composto por três dimensões: pessoas, organizações e tecnologia.

**Figura 13 - Dimensões dos sistemas de informação**



**Fonte: Bruna Diirr, 2021**

- **Pessoas:** Uma empresa, para alcançar o sucesso, é fundamental que as pessoas envolvidas tenham qualificações e treinamentos para efetuar as suas devidas funções;
- **Organizações:** envolve a estrutura e o contexto em que as organizações estão inseridas, refletindo na hierarquia de processos e procedimentos específicos para atender às suas necessidades;
- **Tecnologia:** A ferramenta mais utilizada pelas pessoas a qual tomam decisões e gerentes é a TI para ajudar a enfrentar e superar mudanças.

Conforme Cruz (2000), as dimensões do sistema de informação são simples, facilitando a compreensão de sua importância no funcionamento e aprimoramento dos sistemas de informação.

### **13.5. Trello**

O Trello foi criado para simplificar a organização de projetos, oferecendo um painel que facilita a gestão das atividades a serem feitas. Além disso, a plataforma foi desenvolvida para ter compatibilidade com diversos navegadores móveis, ampliando seu acesso e usabilidade (FLINCO, 2020).

Neste projeto, essa ferramenta foi utilizada para aprimorar o gerenciamento e possibilitar o acompanhamento em tempo real de todos os membros.

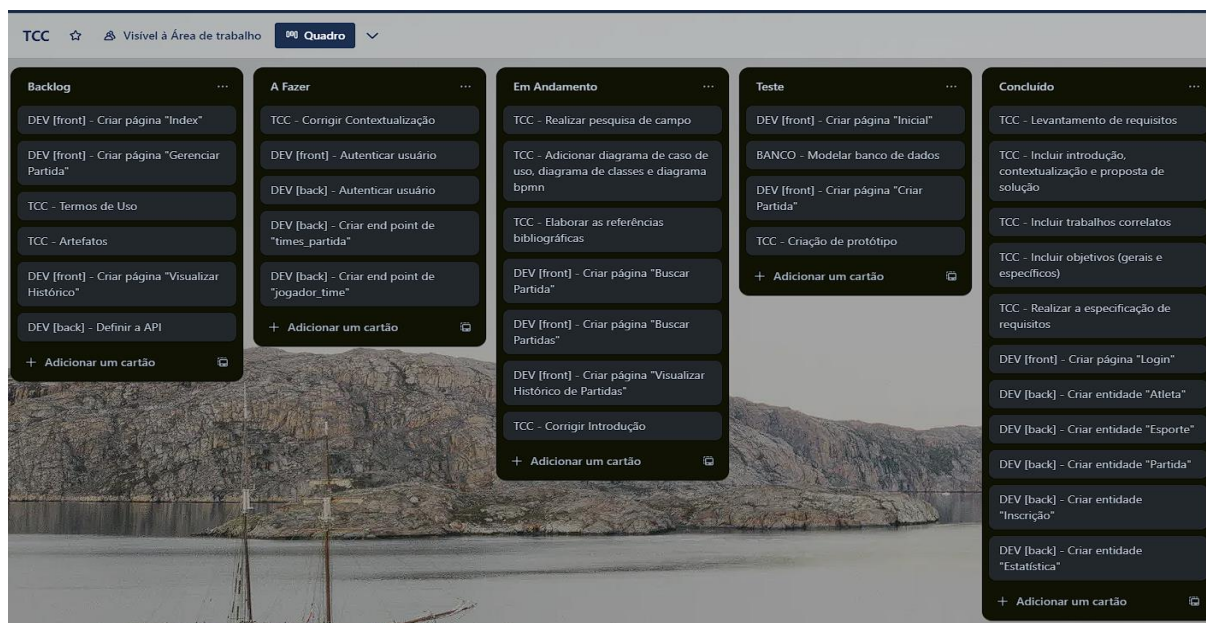
### 13.5.1 Características do Trello

O Trello, desenvolvido por Taiichi Ohmo, utiliza a abordagem Kanban para o controle de atividades. Os projetos são estruturados em quadros que contêm listas de tarefas, representando diversos estágios de progresso, como “Backlog”, “A fazer”, “Em andamento”, “Teste” e “Concluído”. Cada lista reúne cartões que representam tarefas específicas e que podem ser personalizados. Esses cartões podem ser movidos entre os quadros facilitando a visualização do progresso e permitindo uma gestão mais ágil das atividades.

### 13.5.2 Arquitetura do Trello

O criador do Trello, Fog Creek, explica que, em 2011, o *website* foi construído com base no mongoDB, Node.js e Backbone.js (FLINCO, 2020).

**Figura 14 - Trello**



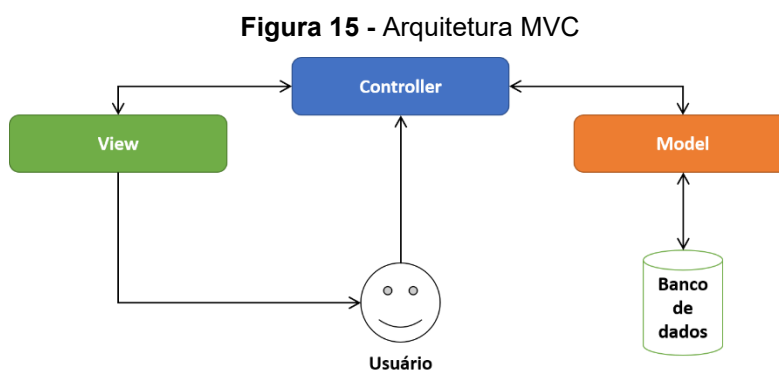
**Fonte: Autores, 2024**

## 14. CARACTERÍSTICAS TÉCNICAS DA SOLUÇÃO

### 14.1. Arquitetura MVC

Segundo Lemos et al. (2013), a arquitetura MVC (*Model-View-Controller*) auxilia os desenvolvedores a construírem aplicações que podem ser reutilizadas em outros projetos, além de facilitar na manutenibilidade do código do sistema. Essa arquitetura é dividida em três camadas: objeto da aplicação e acesso ao banco de dados (*Model*), interface visual pelo usuário (*View*) e processamento das requisições da aplicação (*Controller*).

Para ilustrar melhor essa arquitetura, o diagrama abaixo resume como é a relação entre as três camadas.



Fonte: Café Codificado, 2024

### 14.2. Sistema Web

Segundo Cunha (s.d.), sistema web é um software hospedado na internet que pode ser acessado por qualquer dispositivo com um navegador instalado e uma conexão estável à rede. Essa característica facilita a experiência do usuário por ser desnecessária a instalação do sistema no dispositivo, permitindo que acesse o sistema de qualquer lugar e a qualquer momento.

Além dessa acessibilidade, o sistema web oferece algumas vantagens, como: maior produtividade, custo-benefício, possibilidade de integrações, segurança aprimorada e personalização do sistema (CUNHA, s.d.).

### 14.3. Computação em Nuvem

De acordo com Carissimi (2015), citado em Paula (2021), a ideia da nuvem é oferecer o software, a plataforma e a infraestrutura física como serviços disponíveis por meio da internet.

Segundo Costa (2023), a computação em Nuvem é abordada como:

“um modelo de rede que permite o acesso a um grande conjunto de recursos de computação que são compartilhados, configurados e podem ser solicitados e fornecidos rapidamente com o menor esforço possível por parte do usuário, demandando mínima interação com o provedor de serviços.”

A infraestrutura da nuvem é composta por uma coleção de hardwares, softwares, interfaces, redes de telecomunicação, dispositivos de controle e de armazenamento que permite a entrega da computação na forma de serviço por meio da internet (PEDROSA, 2011). Para que esse tipo de tecnologia seja factível e suporte esse amontoado de dados e sistemas, é necessário que todas as aplicações e dados estejam armazenados em um lugar chamado *data center*. Diante disso, Costa (2023) cita que a computação em nuvem pode ser classificada com cinco características essenciais, três modelos de implantação e três modelos de serviço.

As características essenciais são:

- Serviço *on-demand*: onde o usuário tem acesso aos serviços onde e quando necessário;
- Amplo acesso à rede: os serviços são acessíveis de qualquer plataforma;
- Conjunto agrupado de recursos: mesmo que alocados em diferentes data centers, os recursos são agrupados e oferecidos de forma eficiente;
- Rápida elasticidade: facilidade e liberdade de alocação de maior e menor conjunto de serviços e recursos;
- Serviço mensurável: controle e monitoramento dos serviços pela nuvem.

Os modelos de implantação são:

- Nuvem privada: acessível dentro de um ambiente protegido por firewall;
- Nuvem pública: acessível em toda a internet e à vários usuários simultaneamente, mas separadamente;

- Nuvem híbrida: composta por duas ou mais nuvens que preservem características originais de seu modelo.

Por último, os modelos de serviço:

- Infraestrutura como serviço (IaaS): disponibilização de máquinas virtuais com recursos de processamento, armazenamento, servidores e componentes de rede;
- Plataforma como serviço (PaaS): disponibilização de sistema operacional, linguagens de programação e ambiente de desenvolvimento de aplicações;
- Software como Serviço (SaaS): disponibilização de software, assim o usuário não precisa adquirir licença dele, pagando apenas o que utiliza pelo tempo necessário.

14.4. Quadro Resumo das Ferramentas Utilizadas

- Tecnologias Utilizadas no Back-end.

Quadro 5 - Tecnologias Back-end.

NOME	VERSÃO	VALOR	FABRICANTE
Java	v.17	Gratuita	Oracle
Spring Framework	v.3.3.4	Gratuita	VMware
Maven	v.3.9.8	Gratuita	Apache
Lombok	v.1.18.16	Gratuita	N/A
Intellij Idea	v.2024.2.4	Gratuita	JetBrains
MySQL Community Server	v.8.038	Gratuita	Oracle Corporation

Fonte: Autores, 2024

- Tecnologias Utilizadas no Front-end.

Quadro 6 - Tecnologias Front-end

NOME	VERSÃO	VALOR	FABRICANTE
Angular	v.17	Gratuita	Google
Node.JS	v.20.18.0	Gratuita	OpenJS Foundation
Visual Studio Code	v.1.94.2	Gratuita	Microsoft

Fonte: Autores, 2024

• Outras Ferramentas.

**Quadro 7 - Outras Ferramentas**

NOME	VERSÃO	VALOR	FABRICANTE
Postman	v.11.16.1	Gratuita	Postman, Inc
Figma	N/A	Gratuita	Figma
Trello	N/A	Gratuita	Glitch
Git	v.2.47.0	Gratuita	Git
GitHub	N/A	Gratuita	Microsoft

**Fonte: Autores, 2024**

## 14.5. Ferramentas e Tecnologias Utilizadas no Front-End

### 14.5.1 Angular

De acordo com o a própria desenvolvedora, o Angular é uma estrutura web que facilita a criação de aplicativos rápidos e confiáveis.

“O Angular fornece um amplo conjunto de ferramentas, APIs e bibliotecas para simplificar e agilizar seu fluxo de trabalho de desenvolvimento. Traz uma plataforma sólida para você construir aplicativos rápidos e confiáveis que escalam com o tamanho de sua equipe e o tamanho de sua base de código.” (Angular, s.d.).

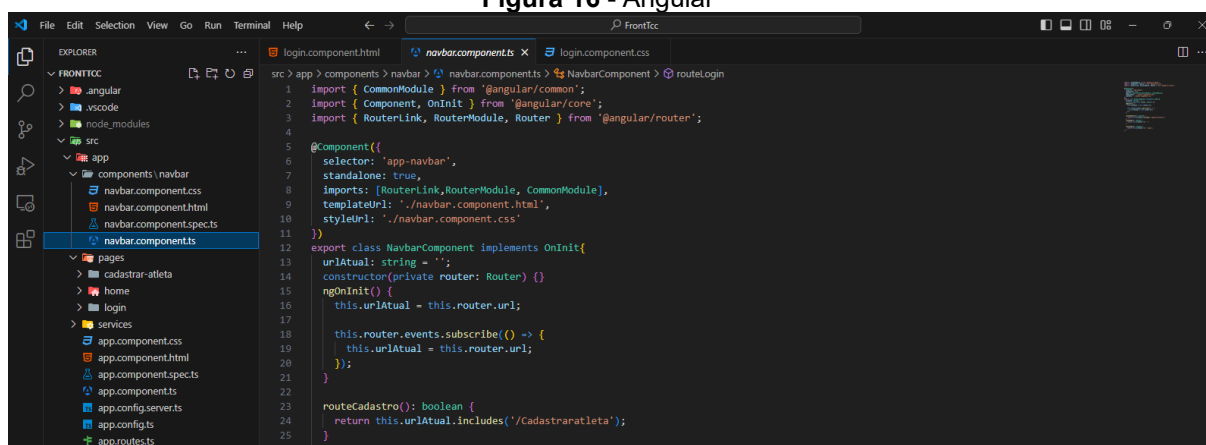
O Angular, criado em 2009 por Miško Hevery e Adam Abrons, é um framework de código aberto estruturado em JavaScript, atualmente mantido e desenvolvido pelo Google. O seu objetivo é obter uma alta produtividade de desenvolvimento, que permite utilizar HTML (HyperText Markup Language) para manipular elementos de uma forma eficiente, usando uma abordagem orientada a componentes para reaproveitamento de linhas de código (BAGLIOTTI,2020).

De acordo com Bagliotti (2020), o Angular possui benefícios, como:

- Reaproveitamento do código por meio de componentes;
- Desempenho aprimorado e um sistema de templates robusto;
- Fornecimento de APIs simplificadas, além de uma depuração facilitada e suporte para carregamento por demandas;
- Compilação rápida do código JavaScript, resultando em melhorias na velocidade de carregamento.

Para essa aplicação, o Angular é utilizado no desenvolvimento do front-end, pela sua capacidade de criar interfaces de usuários dinâmicas e responsivas, além de oferecer ferramentas de integração para a construção de componentes reutilizáveis. A integração do TypeScript ao Angular proporciona uma estrutura mais robusta, o que simplifica futuras manutenções, aumenta a segurança e reduz o número de erros durante o desenvolvimento.

**Figura 16 - Angular**



Fonte: Autores, 2024

### 14.5.2 Node.Js

O Node.Js é um ambiente de execução JavaScript que permite aos desenvolvedores utilizarem essa linguagem tanto no lado do servidor quanto no lado do cliente. Lançado em 2009, o Node.Js possibilitou que o JavaScript fosse executado fora do navegador, trazendo diversas possibilidades de aplicações para o desenvolvimento de soluções back-end utilizando essa linguagem (TILKOV; VINOSKI, 2010).

Entre suas principais características, destaca-se a capacidade de processar várias requisições de maneira simultânea e eficiente, tornando-o uma escolha bastante frequente pelos desenvolvedores que buscam alcançar alto desempenho, estabilidade e eficiência. Com uma comunidade ativa e um ecossistema robusto, o Node.Js tornou-se uma das principais linguagens para o desenvolvimento back-end em JavaScript. (CANTELON, 2014).

Para essa aplicação, o Node.Js fornece um ambiente de execução eficiente e ideal para o desenvolvimento de funcionalidades front-end e back-end de alto desempenho. Sua estrutura permite que o JavaScript seja executado no lado do servidor, simplificando o uso de APIs e facilitando o processamento de requisições.

### **14.5.3 Visual Studio Code**

O Visual Studio Code, popularmente conhecido como “VS Code”, é um editor de código aberto criado pela Microsoft. Essa ferramenta é bastante simples de ser utilizada e possui uma variedade de extensões para ajudar no desenvolvimento (HANASHIRO, 2021). Como editor de código, o VS Code é mais leve do que uma IDE tradicional e vai possuir funcionalidades mais simples. Suas principais características incluem:

- Edição de código com suporte a várias linguagens de programação;
- Terminal de comandos integrado;
- Controle de versão.

Para essa aplicação, o VS Code foi escolhido por sua utilidade nas tarefas de edição de código em múltiplas linguagens, na configuração de arquivos e no desenvolvimento front-end. Além de ser um editor altamente extensível, ele oferece integrações com controladores de versão de códigos, visualizadores de APIs e depuração, e conta com a integração de terminais para o uso de ferramentas, como Node.js diretamente no editor. Diante dessa flexibilidade, o Visual Studio Code torna-se uma ferramenta eficiente e valiosa para o desenvolvimento de projetos.

## **14.6. Ferramentas e Tecnologias Utilizadas no Back-End**

### **14.6.1 IntelliJ Idea**

O IntelliJ IDEA, desenvolvido pela empresa JetBrains, é um ambiente de desenvolvimento integrado (IDE) reconhecido pela sua robustez e pelos recursos avançados para desenvolvimento Java (JETBRAINS, 2020).

Algumas tarefas são demoradas e repetitivas, o que acaba prologando a execução dos projetos e diminuindo a produtividade. Então, com o auxílio do IntelliJ IDEA, o trabalho dos desenvolvedores se torna simplificado e automatizado, pois a ferramenta oferece uma inteligência avançada ao sugerir atalhos e auxílio na escrita de linhas de código em Java (JETBRAINS, 2020).

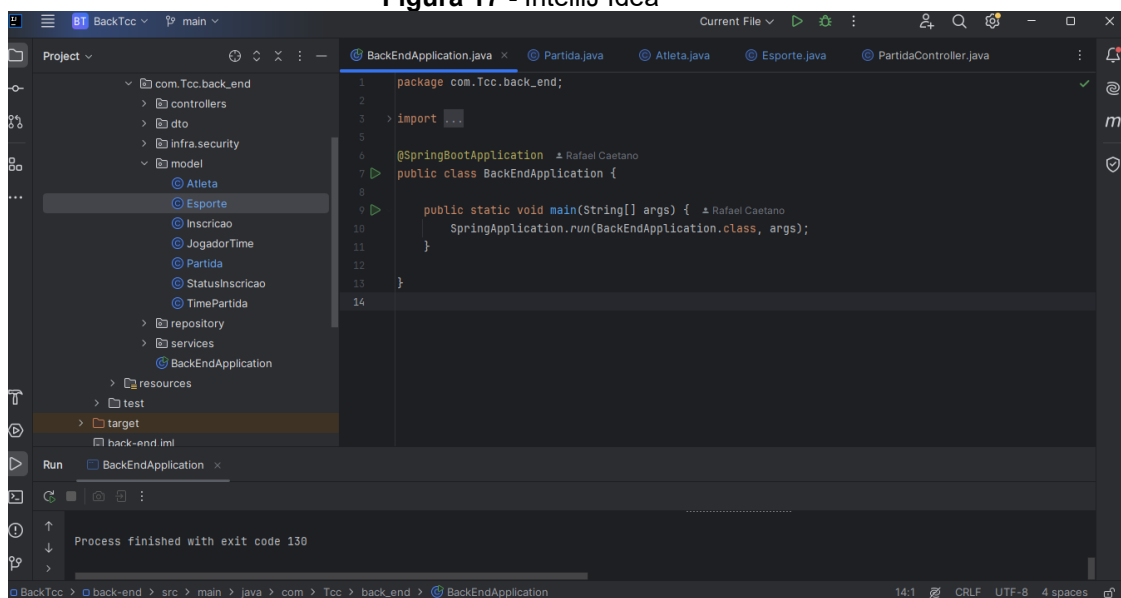
A plataforma não se limita apenas ao desenvolvimento em Java, ela também oferece suporte a outras linguagens como: HTML, SQL, JPQL e JavaScript. Com isso,



há várias possibilidades de o IntelliJ ajudar na praticidade da execução de códigos, como a conclusão instantânea, análise em tempo real e ferramentas avançadas de refatoração.

Para essa aplicação, o IntelliJ foi escolhido como IDE principal para o desenvolvimento dessa aplicação devido às funcionalidades avançadas e ao suporte robusto para tecnologias Java. A ferramenta oferece integração com sistemas de controle de versão, como Git e GitHub, e compatibilidade com o Maven. Com o suporte ativo para frameworks como Spring e Hibernate, o IntelliJ facilita a manutenção do código e o desenvolvimento ágil, proporcionando agilidade às configurações e à execução de testes.

**Figura 17 - IntelliJ Idea**



**Fonte: Autores, 2024**

### 14.6.2 Java

O Java é considerado uma das linguagens de programação mais conhecida e de maior utilização devido a sua portabilidade e robustez. Essa ferramenta segue o princípio do “escreva uma vez, execute em qualquer lugar” que traz a possibilidade que as aplicações feitas em Java seja executada em várias plataformas que suporte o Java Virtual Machine (JVM), sem que seja necessário a adaptação de código.

Conforme citado por Deitel e Deitel (2017), uma das principais características do Java é a estrutura orientada ao objeto, o que possibilita uma melhor estruturação do código e traz a possibilidade de reutilizar componentes de software. Além disso, o Java se mostra uma linguagem altamente tipada, necessitando que o desenvolvedor

declare explicitamente o tipo de dado que deseja armazenar, colaborando assim para a prevenção de erros comumente cometidos na programação.

Outro ponto de destaque é o amplo ecossistema de bibliotecas e frameworks disponíveis, que acelera o desenvolvimento das aplicações proporcionando flexibilidade. Como apontam Sierra e Bates (2014), os frameworks como Hibernate, Struts e Spring são muito utilizados para o desenvolvimento de aplicações web, oferecendo soluções para diversas necessidades.

Para essa aplicação, utilizou-se o Java como linguagem de back-end por conta da sua alta escalabilidade e segurança, características essenciais para servidores web. Horstmann (2019) observa que as APIs nativas de segurança do Java, com suporte à criptografia e autenticação, tornam a linguagem uma escolha de confiança para o desenvolvimento de sistemas que necessitam um bom nível de proteção de dados.

### **14.6.3 Spring Boot**

O Spring Boot é um dos frameworks mais utilizados para o desenvolvimento de aplicações Java, especialmente em aplicações web e sistemas distribuídos. Desenvolvido por Rod Johnson em 2003, o Spring oferece uma infraestrutura robusta e flexível, que facilita a criação de aplicações Java de forma modular e eficiente, abstraindo aspectos técnicos complexos, transações e integrações com outras tecnologias.

De acordo com Walls (2021), o Spring se diferencia através de sua arquitetura baseada nos princípios de inversão de controle (Inversion of Control – IoC) e injeção de dependência (Dependency Injection – DI), que permitem uma clara divisão entre a lógica de negócios e a infraestrutura do sistema. Ao usar o Spring, a responsabilidade de instanciar e gerenciar os objetos para a ser do framework, permitindo ao desenvolvedor se concentrar na lógica da aplicação, tornando o código mais limpo, modular e de fácil manutenção.

Além disso, segundo Sharma (2020), o Spring Framework possui uma grande comunidade de desenvolvedores, permitindo atualizações regulares, novos recursos e correções de segurança, mostrando-se ser uma plataforma confiável para o desenvolvimento de aplicações modernas em Java.

Para essa aplicação, o Spring Boot é utilizado para simplificar o desenvolvimento em Java, com configuração mínima e uma estrutura facilitada. Essa ferramenta fornece um conjunto de padrões que possibilita o processo de criação de aplicativos Spring mais rápido e com maior eficiência. Além disso, ele permite que a aplicação seja executada de maneira independente, reduzindo o tempo do ciclo de desenvolvimento.

#### **14.6.4 Lombok**

O Lombok é uma biblioteca Java com o objetivo de simplificar o desenvolvimento do código, evitando que a escrita seja repetitiva e padronizada, como é a criação dos getters, setters, construtores e outros métodos comuns.

De acordo com Allen (2020), o Lombok é bastante utilizado pelos desenvolvedores Java por ter uma abordagem programática e pela fácil integração com ferramentas de desenvolvimento populares, como IntelliJ IDEA, Eclipse e Maven. Como suas funcionalidades operam na fase de compilação do código, essa ferramenta mostra que não interfere no desempenho e no tempo de execução.

Para essa aplicação, utilizou-se o Lombok para simplificar o código Java, reduzindo a necessidade de escrever manualmente métodos repetitivos como getters e setters. Através de anotações, ele gera automaticamente esses métodos em tempo de compilação, melhorando a legibilidade do código e diminuindo o tempo de desenvolvimento.

#### **14.6.5 Maven**

O Maven é uma ferramenta para automação e gerenciamento de projetos utilizada em projetos em Java. Ele facilita testes, compilações e distribuição de aplicativos, além de ajudar no gerenciamento de dependências de bibliotecas externas.

Com o Maven, configurações do projeto, como dependências, plugins e versões, são centralizadas em somente um arquivo, o “pom.xml”. De acordo com Sonatype (2020), essa centralização permite que o Maven gerencie automaticamente bibliotecas necessárias para o projeto, baixando-as de repositórios remotos e monitorando suas versões. Com essa gestão automatizada, reduz significativamente a dificuldade em configurar manualmente os caminhos das bibliotecas, garantindo a compatibilidades entre elas.

Para essa aplicação, o Maven é utilizado para organizar e simplificar o processo de desenvolvimento. Ele permite definir e centralizar todas as bibliotecas e frameworks necessários em um único arquivo de configuração, possibilitando que essas dependências sejam baixadas e atualizadas automaticamente. Além disso, o Maven facilita a compilação e execução de testes, reduzindo o tempo e o esforço da configuração manual.

#### **14.6.6 Conectividade com Banco de Dados – Hibernate**

O Hibernate é um framework de mapeamento objeto-relacional para Java que simplifica o processo de interação da aplicação Java e o banco de dados. Ele facilita a comunicação entre o modelo de objetos usados durante a construção da aplicação e o modelo do banco de dados relacional, possibilitando que os desenvolvedores manipulem dados como objetos Java, evitando a necessidade de fazer consultas SQL complexas manualmente.

Conforme Bauer e King (2015), o maior benefício do Hibernate é o mapeamento automático das classes Java para as tabelas do banco de dados através de arquivos XML. Com isso, as classes de entidade, seus atributos e os relacionamentos podem ser inseridos diretamente em tabelas relacionas, facilitando a manipulação de dados de forma orientada a objetos.

Para essa aplicação, o framework Hibernate foi escolhido para facilitar a interação entre o código Java e o banco de dados MySQL. Através dele, é possível fazer inserção, consulta, atualização e exclusão de forma simplificada, pois o Hibernate converte automaticamente os dados entre objetos Java e as tabelas do banco de dados. Além disso, o framework disponibiliza recursos de controle de transações e cache, otimizando o desempenho e garantindo a integridade dos dados da aplicação.

#### **14.6.7 Segurança nas suas Aplicações - Spring Security**

O Spring Security é um framework robusto e amplamente utilizado para garantir segurança nas aplicações inspiradas no ecossistema Spring. Ele oferece a autenticação e o controle de acesso detalhado para a proteção de aplicativos web e

APIs, possibilitando a implementação de políticas de segurança flexíveis e personalizáveis.

De acordo com Walls (2021), o Spring Security disponibiliza mecanismos para a verificação da identidade do usuário e do controle de acesso a recursos com base em cargos. Ele tem suporte para diferentes métodos de autenticação, como formulários, autenticação HTTP básica, OpenID Connect, OAuth2, e até possibilita a integração com fornecedores de identidade externas, como Google e Facebook.

Para essa aplicação, o Spring Security foi adotado para implementar recursos de segurança bem estruturados, alcançando proteção de dados e controle de acesso aos recursos da API. Ele permite implementar medidas de segurança avançada de forma simplificada, possibilitando a proteção das informações do usuário e a integridade do sistema. Assim, a aplicação torna-se confiável e segura contra acessos não autorizados.

#### **14.6.8 Postman**

O Postman é uma ferramenta comumente utilizada para o desenvolvimento, testes e documentação de APIs, facilitando o envio de requisições HTTP e a simulação de diversos cenários de interação com APIs, simplificando assim o processo de validação. Essa ferramenta conta com uma interface gráfica intuitiva, onde podem ser criadas e organizadas requisições HTTP incluindo métodos como GET, POST, PUT e DELETE (KULKARNI, 2020). As requisições podem incluir alguns parâmetros, corpos de mensagens e cabeçalhos, mostrando-se ideal para o teste de APIs.

Para essa aplicação, o Postman é utilizado para testar e validar requisições da API de forma simplificada e eficiente. Ele facilita a criação, o envio e a análise de requisições, além de permitir que seja verificado o funcionamento de endpoints através de testes em diversos cenários e respostas do sistema. Essa ferramenta também simula autenticações, testa parâmetros e analisa o retorno de dados, permitindo uma identificação rápida de problemas.

#### **14.6.9 Git**

O Git é um sistema para controle de versão distribuída e possui grande uso para o gerenciamento de código fonte em projetos de desenvolvimento de software. Esse sistema facilita o rastreamento de mudança de código, possibilitando que vários

desenvolvedores trabalhem em sintonia no mesmo projeto, sem conflito de versões ou perda de dados.

Uma das fortes características do Git é a sua capacidade de gerenciar ramificações, possibilitando ao desenvolvedor a criação de uma ramificação separada do código fonte, o que facilita o trabalho com a implementação de novas funções ou até correção de bugs. Ao final, essa ramificação pode ser mesclada ao código fonte, mantendo assim o histórico de alterações, facilitando o controle de versões.

Para essa aplicação, o Git foi utilizado como sistema de controle de versão de código, permitindo gerenciar e acompanhar o desenvolvimento de forma eficiente e organizada. Com isso, o sistema facilita o rastreamento de mudanças, correção de erros e colaboração entre a equipe, garantindo a integridade do código ao longo do projeto.

#### **14.6.10 GitHub**

O GitHub, uma plataforma baseada no sistema de controle de versão Git, oferece ferramentas de hospedagem, colaboração e gerenciamento de repositórios de códigos fontes. Após seu lançamento em 2008, o GitHub se tornou rapidamente uma das principais plataformas para desenvolvedores compartilharem projetos e montarem portfólios, além de ser usado para projetos privados das empresas (GITHUB, 2023).

Para essa aplicação, o GitHub foi escolhido como plataforma de armazenamento de repositórios e controle de versão de código, possibilitando assim que toda a equipe acompanhe, colabore e revise as mudanças feitas no sistema de forma organizada e segura. Além disso, possibilita a visualização do histórico de alterações, a reversão de modificações e o trabalho em diferentes *branches* para desenvolver novas funcionalidades, sem comprometer o código principal.

#### **14.6.11 API**

A Interface de Programação de Aplicações, em inglês *Application Programming Interface*, mais conhecido como API é um conjunto de definições que permitem que vários softwares façam a comunicação entre si. As APIs são como pontes, permitindo

que as aplicações acessem funcionalidades ou dados de outros sistemas de forma estruturada e segura.

Essas interfaces são bastante utilizadas para vários contextos de desenvolvimento de *software*, desde aplicações *web* até aplicativos *mobile*. Elas entregam a possibilidade de diferentes componentes de *software*, independente da linguagem de programação ou plataforma, interagirem de maneira padronizada, reduzindo a necessidade de reescrever funções já existentes, o que promove a reutilização de dados.

Para essa aplicação, a utilização da API foi de extrema importância para permitir a comunicação entre o front-end e o back-end de forma estruturada, garantindo uma troca de dados segura e eficiente. Com a API, foi possível modularizar funcionalidades, entregando *endpoints* específicos para cada ação, facilitando a escalabilidade do sistema e futuras integrações com outras plataformas.

No quadro abaixo está exibindo as operações disponíveis da api do sistema Resenha Esporte Clube.

**Quadro 8 - Endpoints API**

Auth		
POST	register	Permite cadastrar o jogador no sistema.
POST	login	Permite autenticar o usuário.
Atleta		
GET	retornarAtletas	Consulta todos os atletas registrados.
GET	buscarAtletaPorId	Consulta o atleta pelo id informado.
PUT	atualizarAtleta	Atualiza os dados do atleta de acordo com o objeto atleta passado.
Esportes		
GET	retornarEsportes	Consulta todos os esportes registrados.

GET	buscarEsportePorId	Consulta o esporte de acordo com id informado.
POST	novoEsporte	Permite cadastrar um esporte no sistema.
PUT	atualizarEsporte	Atualiza os dados do esporte de acordo com o objeto esporte passado.
<b>Gerenciar</b>		
POST	salvarTimes	Permite cadastrar os times no sistema.
GET	retornarTimesIdPartida	Consulta os times de acordo com id da partida informado.
PUT	atualizarTimes	Permite atualizar as informações de cada time.
<b>Inscricao</b>		
GET	retornarInscricao	Consulta todas as inscrições registradas no sistema.
GET	retornarInscricaoPorId	Consulta a inscrição de acordo com o id informado.
GET	buscarInscricaoPorIdPartida	Consulta todas as inscrições de um acordo com o id informado da partida.
POST	novalInscricao	Permite cadastrar uma nova inscrição em uma partida.
PUT	cancelarInscricao	Permite o atleta cancelar sua inscrição em uma partida.
PUT	removerInscrito	Permite o criador do atleta remover um inscrito da sua partida.
<b>Partida</b>		
GET	buscarPartidaPorId	Consulta a partida de acordo com id informado.



GET	retornarPartidas	Consulta as partidas que não foram criadas pelo atleta de acordo com o id do atleta informado e onde o id do status da partida é 1.
GET	getPartidasByInscricaoId	Consulta as partidas que o atleta está inscrito de acordo com id do atleta informado.
GET	getPartidasByAtletaid	Consulta as partidas criadas pelo atleta de acordo com id do atleta informado.
POST	novaPartida	Permite cadastrar uma partida no sistema.
PUT	atualizarPartida	Permite atualizar os dados da partida de acordo com objeto partida passado.
PUT	cancelarPartida	Permite o atleta cancelar a sua partida.
PUT	finalizarPartida	Permite o atleta finalizar a sua partida.
<b>StatusInscricao</b>		
GET	retornarStatusInscricao	Consulta os status das inscrições.
GET	buscarStatusInscricaoPorId	Consulta o status da inscrição de acordo com id informado.
POST	novoStatusInscricao	Permite cadastrar uma novo status inscrição no sistema.
<b>StatusPartida</b>		
GET	retornarStatusPartida	Consulta os status das partidas.
GET	buscarStatusPartidaPorId	Consulta o status da partida de acordo com id informado.
PUT	novoStatusPartida	Permite cadastrar uma novo status partida no sistema.

#### **14.6.12 MySQL**

O MySQL é um sistema de gerenciamento de banco de dados relacional, bastante usado em aplicações web por conta da sua segurança, desempenho e flexibilidade. Projetado para processar grandes volumes de dados, o MySQL disponibiliza o suporte para ações de consulta, inserção, atualização e exclusão de forma simplificada e eficiente.

Para essa aplicação, o MySQL foi escolhido por conter uma grande capacidade de manipular dados estruturados com segurança e pela facilidade de integração com a linguagem Java, utilizada no back-end da aplicação. Além disso, ele possui ferramentas estruturadas para o gerenciamento de dados e controle de transações, o que garante a integridade dos dados para operações complexas, bem como a simplicidade na manutenção e na escalabilidade da aplicação, tornando-o uma solução sustentável e eficiente para o armazenamento de dados.

#### **14.6.13 Draw.io**

O Draw.io é uma ferramenta de design gráfico para a criação de diagramas, fluxograma e outros tipos de representações visuais, sendo utilizado especialmente em projetos de desenvolvimento de software. Ele é essencial para representar arquiteturas, fluxo de trabalho e processos complexos de forma visual e organizada. Com uma interface intuitiva e recursos de arrastar e soltar, a ferramenta simplifica a criação de diversos tipos de diagramas, que são necessários para a documentação e a análise de requisitos de projetos de software.

Para essa aplicação, a escolha do Draw.io se justifica pela facilidade que a ferramenta permite na criação e compartilhamento de diagramas, bem como nos projetos de colaboração em equipe. Além disso, a ferramenta é gratuita, acessível e não compromete a qualidade das representações gráficas.

### **15. ESPECIFICAÇÃO DE REQUISITOS**

De acordo com Cunha (s.d.), a engenharia de requisitos tem como objetivo gerar um produto que é uma documentação de requisitos ou mesmo uma criação de protótipos por meio de um conjunto de tarefas a serem executadas. Com isso, também

afirma que os requisitos funcionais são todos os problemas e necessidades que o software deve resolver, seja por meio de funções ou serviços.

Já os requisitos não funcionais são todos aqueles referentes ao software ou ao *app* que tornará aquilo que está sendo projetado em algo concreto. Ou seja, os requisitos funcionais estão focados no que será feito, enquanto os não funcionais descrevem como será feito.

No quadro 9 está representado os requisitos funcionais do sistema Resenha Esporte Clube.

**Quadro 9 - Requisitos Funcionais**

<b>Código</b>	<b>Requisito</b>	<b>Descrição</b>	<b>Prioridade</b>
RF 01	Manter Jogador	Cadastrar jogadores para um maior controle de quem utiliza o sistema.	1
RF 02	Realizar Login	Efetuar o login para ter acesso ao sistema e as funcionalidades dele.	2
RF 03	Criar Partidas	Permitir criar listas para que jogadores se inscrevam para participar de partidas de esportes variados cadastrados no sistema.	3
RF 04	Gerenciar Partidas	Permitir que o jogador que criou a partida consiga gerenciar a mesma, cancelando e marcando as estatísticas das partidas.	4
RF 05	Buscar Partidas	Permitir que os jogadores busquem partidas do esporte favorito.	5
RF 06	Inscrever-se na Partida	Permitir que jogadores se inscrevam em partidas de esportes variados para participar das mesmas.	6
RF 07	Gerar Estatísticas	Gerar estatísticas ao fim do jogo.	7
RF 08	Sortear Times	Permitir que o jogador que criou a lista sorteie os times aleatoriamente com os jogadores inscritos.	8
RF 09	Cronometrar	Disponibilizar um cronômetro para um controle de tempo das partidas.	9

RF 10	Visualizar Histórico de Partidas	Permitir que o jogador consiga visualizar todas as partidas que participou.	10
RF 11	Visualizar Estatísticas	Após o término das partidas, permite ao jogador visualizar as estatísticas geradas conforme o somatório das pontuações das vitórias, derrotas e empates.	11

**Fonte: Autores, 2024**

No quadro 9 está representado os requisitos não funcionais do sistema Resenha Esporte Clube.

**Quadro 10 - Requisitos Não Funcionais**

Código	Requisito	Descrição
RNF 01	Segurança	Garantir que todas as informações pessoais e sensíveis estejam protegidas e seguras contra acesso não autorizado.
RNF 02	Confiabilidade	Garantir que a plataforma esteja sempre disponível e não apresenta interrupções frequentes.
RNF 03	Usabilidade	Garantir que a plataforma seja fácil de usar e acessível para usuários de todas as idades e níveis de habilidade.
RNF 04	Escalabilidade	Garantir que a plataforma possa lidar com muitos usuários e crescer à medida que a comunidade cresce.

**Fonte: Autores, 2024**

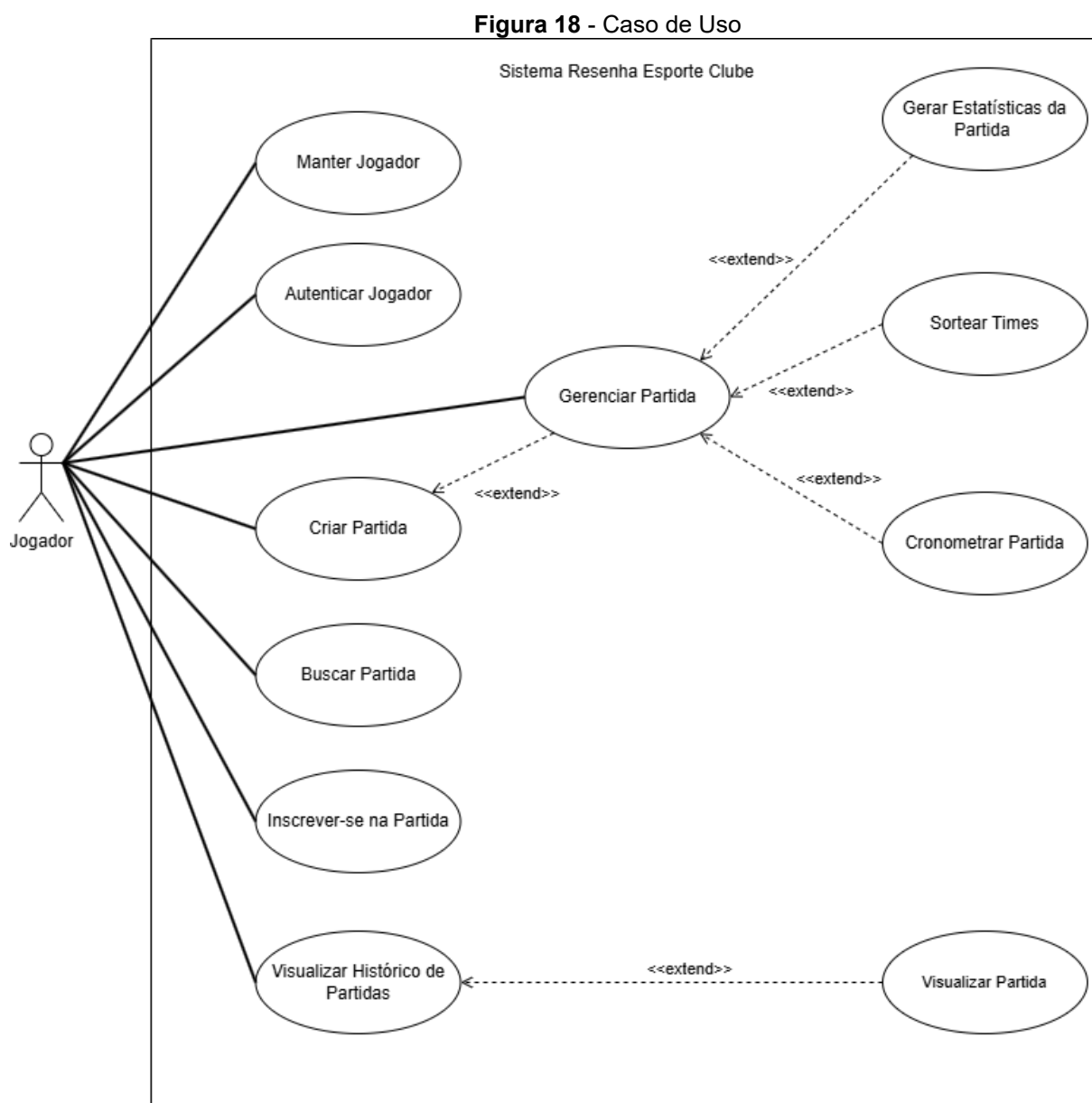
## 16. CASO DE USO

### 16.1. Diagrama de Caso de Uso

De acordo com Guedes (2018), o objetivo do diagrama de caso de uso é permitir que o usuário tenha uma visualização simplificada das funcionalidades que o sistema deve implementar. Esse diagrama possibilita a compreensão acessível sobre o comportamento do sistema, mesmo para aqueles que tenham conhecimento básico do problema identificado, permitindo uma apresentação na perspectiva dos usuários.

Além disso, o diagrama também busca identificar os possíveis tipos de usuários que utilizarão o sistema, definir quais os papéis que eles assumirão e

especificar as funcionalidades específicas para cada perfil dentro da plataforma (GUEDES, 2018).



Fonte: Autores, 2024

## 16.2. Especificação dos Casos de Usos

De acordo com Guedes (2018), a especificação de casos de usos tem como intuito descrever, por meio de uma linguagem de fácil entendimento, informações como:

- Qual a função que aquele caso de uso específico pode realizar;
- Quais os atores que interagem com ele;
- Etapas que devem ser realizadas pelo ator e pelo sistema para que execute a sua função/tarefa do caso de uso;

- Quais as restrições esse caso de uso tem;
- Quais parâmetros devem ser inseridos;
- Validações que deve possuir.

Nos itens abaixo serão detalhados cada caso de uso paralelamente com o protótipo.

### 16.2.1 Manter Jogador

Neste caso de uso onde o ator principal vai ser o jogador, deve possibilitar o cadastro do jogador no sistema.

#### 16.2.1.1 Detalhamento de Caso de Uso

**Quadro 11 - Especificação de Caso de Uso Manter Jogador**

<b>Nome do Caso de Uso</b>	Manter Jogador.
<b>Atores Principais</b>	Jogador.
<b>Atores Secundários</b>	Sistema.
<b>Resumo</b>	Este caso de uso descreve o processo de cadastro do usuário no sistema.
<b>Pré-Condições</b>	Acessar o sistema. Não ter cadastro no sistema. O jogador deve possuir dados válidos para cadastro.
<b>Pós-Condições</b>	O jogador conseguirá acessar o sistema e suas funcionalidades.
<b>Fluxo Principal</b>	
<ol style="list-style-type: none"> <li>1. O jogador deverá acessar a página de cadastro</li> <li>2. O jogador insere os dados no formulário.</li> <li>3. O jogador aciona o botão “cadastrar”.</li> <li>4. O sistema valida se os dados estão corretos.</li> <li>5. O sistema salva os dados fornecidos na base de dados.</li> <li>6. O jogador está apto a fazer o login no sistema.</li> <li>7. O sistema redireciona o usuário para a página de login.</li> </ol>	

8. Fim do caso de uso.
<b>Fluxo Alternativo I - Desistir do Cadastro</b>
<ol style="list-style-type: none"> <li>1. O jogador aciona o botão “Cancelar”.</li> <li>2. O caso de uso se encerra.</li> </ol>
<b>Fluxo de Exceção – Dados do Formulário de Cadastro Incorretos</b>
<ol style="list-style-type: none"> <li>1. O sistema valida se todos os campos foram preenchidos.</li> <li>2. O sistema exibe uma mensagem avisando qual o campo que não está preenchido.</li> <li>3. O jogador tenta novamente inserindo os dados nos campos não preenchidos.</li> <li>4. O sistema pode estar fora do ar.</li> <li>5. O caso de uso se encerra.</li> </ol>

Fonte: Autores, 2024

#### 16.2.1.2 Protótipo

A tela a seguir representa a tela de cadastro do jogador, que permite com que ele se registre no sistema preenchendo os campos fornecidos.

Figura 19 - Protótipo Manter Jogador

Fonte: Autores, 2024

#### 16.2.2 Autenticar Jogador

Neste caso de uso onde o ator principal é o jogador, deve possibilitar este o acesso à aplicação inserindo as suas credenciais de login, que seria o e-mail e senha.

### 16.2.2.1 Detalhamento de Caso de Uso

**Quadro 12 - Especificação de Caso de Uso Autenticar Jogador**

<b>Nome do Caso de Uso</b>	Autenticar Jogador.
<b>Atores Principais</b>	Jogador
<b>Atores Secundários</b>	Sistema.
<b>Resumo</b>	Este caso de uso possibilita o jogador acessar o sistema caso ele já tenha um cadastro ativo.
<b>Pré-Condições</b>	Acessar o sistema. O jogador deve possuir cadastro ativo no sistema.
<b>Pós-Condições</b>	O jogador estará logado na aplicação e conseguirá acessar as funcionalidades do sistema.
<b>Fluxo Principal</b>	
<ol style="list-style-type: none"> <li>1. O jogador acessa a tela de login da aplicação.</li> <li>2. O jogador informa os dados de usuário e senha.</li> <li>3. O sistema verifica a veracidade dos dados.</li> <li>4. O sistema autentica o usuário.</li> <li>5. O sistema permite o acesso à aplicação.</li> <li>6. O sistema redireciona o usuário para a página principal.</li> <li>7. O caso de uso se encerra.</li> </ol>	
<b>Fluxo de Exceção – Dados de <i>Login</i> Inválidos</b>	
<ol style="list-style-type: none"> <li>1. O sistema verifica que as informações de autenticação estão incorretas.</li> <li>2. O sistema exibe uma mensagem de erro informando que os dados de autenticação estão incorretos.</li> <li>3. O jogador pode tentar novamente, inserindo as informações corretas.</li> <li>4. O sistema pode estar fora do ar.</li> <li>5. O caso de uso se encerra.</li> </ol>	

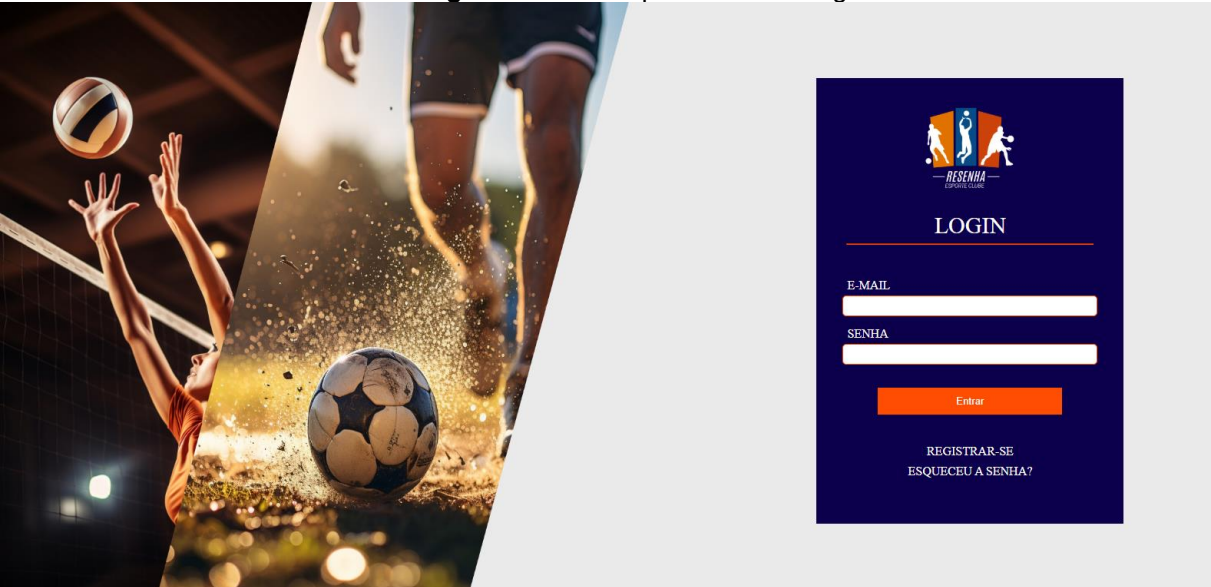
**Fonte: Autores, 2024**



16.2.2.2 Protótipo

A tela a seguir permite que o jogador acesse o sistema inserindo o seu e-mail e senha.

Figura 20 - Protótipo Autenticar Jogador



Fonte: Autores, 2024

16.2.3 Criar Partida

Neste caso de uso o ator principal vai ser o jogador, e permite que ele crie suas próprias partidas.

16.2.3.1 Detalhamento de Caso de Uso

Quadro 13 - Especificação de Caso de Uso Criar Partida

Nome do Caso de Uso	Criar Partida.
Atores Principais	Jogador.
Atores Secundários	Sistema.
Resumo	Este caso de uso descreve o processo de criação de partida.
Pré-Condições	Ter executado caso de uso “Autenticar jogador”. O jogador deve possuir cadastro ativo no sistema.

<b>Pós-Condições</b>	A partida será disponibilizada no sistema para que outros jogadores se inscrevam.
<b>Fluxo Principal</b>	
<ol style="list-style-type: none"> <li>1. O jogador acessa o sistema.</li> <li>2. O jogador, ao acessar a página principal, seleciona a opção “Criar Partida do Meu Esporte Favorito”.</li> <li>3. O sistema redireciona o atleta para uma página com um formulário.</li> <li>4. O jogador preenche os campos do formulário: nome da partida, esporte a ser praticado, quantidade máxima de atletas, data, horário de início, horário de término, nome do local, endereço, descrição e, por fim, selecionar uma região administrativa que será realizada a partida.</li> <li>5. O jogador aciona o botão “Cadastrar”.</li> <li>6. O sistema salva os dados na base de dados.</li> <li>7. O sistema direciona o jogador para a página “Gerenciar Partida”.</li> <li>8. O caso de uso se encerra.</li> </ol>	
<b>Fluxo Alternativo I – Desistir de criar partida</b>	
<ol style="list-style-type: none"> <li>1. O jogador seleciona a opção “Cancelar”.</li> <li>2. O caso de uso se encerra.</li> </ol>	
<b>Fluxo de Exceção – Dados do formulário inválidos</b>	
<ol style="list-style-type: none"> <li>1. O sistema verifica se todos os campos foram preenchidos.</li> <li>2. O sistema notifica os campos não preenchidos corretamente.</li> <li>3. O sistema valida se o horário de término é maior que o horário de início.</li> <li>4. O sistema pode informar que o horário de término pode não pode ser registrado.</li> <li>5. O sistema pode estar fora do ar.</li> <li>6. O caso de uso se encerra.</li> </ol>	

Fonte: Autores, 2024

### 16.2.3.2 Protótipo

A tela a seguir permite que o usuário acesse a página de criar partida.

Figura 21 - Protótipo Criar Partida

Preencha os campos do formulário abaixo.

### Criar Partida

Nome da Partida

Data

Nome do Local

Esporte

Horário de Início

Endereço

Selecione uma Região Administrativa:

Quantidade de atletas

Horário de Término

Descrição

**Campos Obrigatórios \***

Cancelar Cadastrar

Fonte: Autores, 2024

### 16.2.4 Buscar Partida

Neste caso de uso onde o ator principal é o jogador, deve permitir que os usuários busquem por partidas dentro do sistema.

#### 16.2.4.1 Detalhamento do Caso de Uso

Quadro 14- Especificação de Caso de Uso Buscar Partida

<b>Nome do Caso de Uso</b>	Buscar Partida.
<b>Atores Principais</b>	Jogador.
<b>Atores Secundários</b>	Sistema.
<b>Resumo</b>	Este caso de uso possibilita o jogador buscar partidas dentro do sistema.
<b>Pré-Condições</b>	O jogador deve ter executado o caso de uso "Autenticar Jogador". O jogador deve possuir cadastro ativo no sistema. O sistema ter partidas cadastradas.
<b>Pós-Condições</b>	N/A.
<b>Fluxo Principal</b>	

1. O jogador acessa o sistema.
2. O jogador, ao acessar a página principal, seleciona a opção “Buscar partidas próximas a mim”.
3. O sistema exibe as partidas registradas no sistema.
4. O jogador pode buscar a partida que deseja pelo nome da partida no campo “Buscar”.
5. O jogador pode filtrar o esporte que deseja visualizar selecionando o “Esporte”.
6. O jogador pode filtrar a região administrativa que deseja visualizar selecionando “RA”.
7. O sistema retorna a partida de acordo com o filtro ou pesquisa.
8. O jogador seleciona a partida que deseja visualizar.
9. O sistema direciona o jogador para a modal “Inscrever-se na partida”.
10. O caso de uso se encerra.

#### **Fluxo Alternativo I – Inscrever-se na partida**

1. O sistema executa o caso de uso “Inscrever-se na Partida”.
2. O caso de uso se encerra.

#### **Fluxo de Exceção – Inexistência de partidas**

1. O sistema pode não conter partidas cadastradas.
2. O sistema pode não conter partidas com o filtro selecionado.
3. O sistema pode não conter partidas com o nome fornecido na busca.
4. O sistema pode estar fora do ar.
5. O caso de uso se encerra.

**Fonte: Autores, 2024**

#### **16.2.4.2 Protótipo**

A tela a seguir permite a visualização de todas as partidas ofertadas no sistema para que o jogador possa encontrar a partida adequada para ele.

Figura 22 - Protótipo Buscar Partida



Fonte: Autores, 2024

16.2.5 Inscrever-se na Partida

Neste caso de uso onde o ator principal é o jogador, deve possibilitar a inscrição dos usuários nas partidas selecionadas.

16.2.5.1 Detalhamento do Caso de Uso

Quadro 15 - Especificação de Caso de Uso Inscrever-se na Partida

Nome do Caso de Uso	Inscrever-se na Partida.
Atores Principais	Jogador.
Atores Secundários	Sistema.
Resumo	Este caso de uso possibilita que o jogador se inscreva em uma partida.
Pré-Condições	O jogador deve ter executado o caso de uso “Autenticar Jogador”. O jogador deve possuir cadastro ativo no sistema. O sistema possuir partidas cadastradas. O jogador ter executado o caso de uso “Buscar Partidas”.
Pós-Condições	O jogador é inscrito na partida selecionada.
Fluxo Principal	

1. O jogador acessa o sistema.
2. O jogador busca partidas.
3. O jogador seleciona a partida que deseja se inscrever para participar.
4. O sistema expande o *card* da vaga selecionada possibilitando que o jogador se inscreva.
5. O jogador seleciona a opção “Inscrever-se”.
6. O sistema inscreve o jogador na base de dados na partida selecionada.
7. O sistema o redireciona para a tela de “Visualizar Partida”.
8. O caso de uso se encerra.

#### Fluxo Alternativo I - Desistir de realizar inscrição

1. O jogador seleciona o botão “Fechar”.
2. O caso de uso se encerra.

#### Fluxo de Exceção – Indisponibilidade de realizar inscrição

1. A partida pode estar com o número máximo de inscrições.
2. A partida pode ter sido cancelada.
3. O sistema pode estar fora do ar.

Fonte: Autores, 2024

### 16.2.5.2 Protótipo

A tela a seguir permite a visualização das informações gerais de uma partida, assim como a possibilidade do usuário se inscrever na partida selecionada.

Figura 23 - Protótipo Inscrever-se na Partida



Fonte: Autores, 2024

### 16.2.6 Visualizar Histórico de Partidas

Neste caso de uso onde o ator principal é o jogador, deve possibilitar a visualização de todas as partidas que o jogador está inscrito, abrangendo diferentes status, como cancelada, finalizada e ativa.

#### 16.2.6.1 Detalhamento do Caso de Uso

**Quadro 16 - Especificação de Caso de Uso Visualizar Histórico de Partidas**

<b>Nome do Caso de Uso</b>	Visualizar Histórico de Partidas.
<b>Atores Principais</b>	Jogador.
<b>Atores Secundários</b>	Sistema.
<b>Resumo</b>	Este caso de uso possibilita ao jogador visualizar todas as partidas nas quais está inscrito, abrangendo diferentes status, como cancelada, finalizada e ativa.
<b>Pré-Condições</b>	<p>O jogador deve ter executado o caso de uso “Autenticar Jogador”.</p> <p>O jogador deve possuir cadastro ativo no sistema.</p> <p>O jogador deve ter se inscrito em pelo menos uma partida no sistema.</p>
<b>Pós-Condições</b>	N/A.
<b>Fluxo Principal</b>	
<ol style="list-style-type: none"> <li>1. O jogador acessa o sistema.</li> <li>2. O jogador, no <i>sidebar</i> a esquerda, seleciona a opção “Visualizar Histórico”.</li> <li>3. O sistema mostra todas as partidas que o jogador se inscreveu.</li> <li>4. O jogador visualiza as partidas com suas principais informações.</li> <li>5. O jogador aciona o botão “Ver Partida”.</li> <li>6. O sistema redireciona o jogador para a tela de “Visualizar Partida”.</li> <li>7. O caso de uso se encerra.</li> </ol>	
<b>Fluxo de Exceção</b>	
<ol style="list-style-type: none"> <li>1. O atleta pode não ter se inscrito em pelo menos uma partida.</li> <li>2. A partida pode ter sido cancelada.</li> <li>3. O sistema pode estar fora do ar.</li> </ol>	

**Fonte: Autores, 2024**

16.2.6.2 Protótipo

A tela a seguir permite a visualização de todas as partidas que o usuário está inscrito.

Figura 24 - Protótipo Visualizar Histórico de Partidas



Fonte: Autores, 2024

16.2.7 Visualizar Partida

Neste caso de uso onde o ator principal é o jogador, deve possibilitar a visualização das informações da partida selecionada.

16.2.7.1 Detalhamento do Caso de Uso

Quadro 17 - Especificação de Caso de Uso Visualizar Partida

Nome do Caso de Uso	Visualizar partida.
Atores Principais	Jogador.
Atores Secundários	Sistema.
Resumo	Este caso de uso possibilita ao jogador visualizar as inscrições e informações da partida.



<b>Pré-Condições</b>	<p>O jogador deve ter executado o caso de uso “Autenticar Jogador”.</p> <p>O jogador deve possuir cadastro ativo no sistema.</p> <p>O jogador deve ter executado o caso de uso “Visualizar histórico de partida”</p>
<b>Pós-Condições</b>	N/A.
<b>Fluxo Principal</b>	
<ol style="list-style-type: none"> <li>1. O sistema exibe as informações e inscrições da partida selecionada.</li> <li>2. O caso de uso se encerra.</li> </ol>	
<b>Fluxo Alternativo I - Cancelar Inscrição</b>	
<ol style="list-style-type: none"> <li>1. O jogador aciona o botão “Cancelar Inscrição”.</li> <li>2. O sistema solicita uma confirmação.</li> <li>3. O jogador confirma que deseja cancelar a inscrição.</li> <li>4. O sistema realiza o cancelamento.</li> <li>5. O sistema redireciona o jogador para a tela “Visualizar Histórico”.</li> </ol>	
<b>Fluxo de Exceção</b>	
<ol style="list-style-type: none"> <li>1. A partida pode ter sido cancelada.</li> <li>2. O sistema pode estar fora do ar.</li> <li>3. O caso de uso se encerra.</li> </ol>	

Fonte: Autores, 2024

### 16.2.7.2 Protótipo

A tela a seguir permite a visualização das informações detalhadas de uma partida.

Figura 25 – Protótipo Visualizar Partida



Fonte: Autores, 2024

### 16.2.8 Gerenciar Partida

Neste caso de uso onde o ator principal é o jogador, deve possibilitar o gerenciamento das partidas criadas por ele.

#### 16.2.8.1 Detalhamento do Caso de Uso

**Quadro 18 - Especificação de Caso de Uso Gerenciar Partida**

<b>Nome do Caso de Uso</b>	Gerenciar partida.
<b>Atores Principais</b>	Jogador.
<b>Atores Secundários</b>	Sistema.
<b>Resumo</b>	Esse caso de uso permite que o organizador da partida edite as informações, cancele o evento e visualize as inscrições.
<b>Pré-Condições</b>	O jogador deve ter executado o caso de uso "Autenticar Jogador". O jogador deve possuir cadastro ativo no sistema. O jogador deve ter executado o caso de uso "Criar Partida".
<b>Pós-Condições</b>	N/A.
<b>Fluxo Principal</b>	
<ol style="list-style-type: none"> <li>1. O jogador acessa o sistema.</li> <li>2. O jogador, no <i>sidebar</i> a esquerda, seleciona a opção "Gerenciar partidas".</li> <li>3. O sistema redireciona o jogador para a página "Minhas Partidas".</li> <li>4. O sistema lista as partidas criadas pelo jogador.</li> <li>5. O jogador seleciona a partida que deseja gerenciar acionando o botão "Gerenciar".</li> <li>6. O sistema redireciona o jogador para a página "Gerenciar partida".</li> <li>7. O sistema exibe as informações da partida.</li> <li>8. O caso de uso se encerra.</li> </ol>	
<b>Fluxo Alternativo I – Opções de funcionalidades</b>	
<ol style="list-style-type: none"> <li>1. O jogador pode executar os seguintes casos de uso: <ol style="list-style-type: none"> <li>1.1. Sortear Times;</li> <li>1.2. Cronometrar Partida;</li> <li>1.3. Gerar Estatísticas da Partida.</li> </ol> </li> <li>2. O caso de uso se encerra.</li> </ol>	

### Fluxo de Exceção – Cancelamento da partida

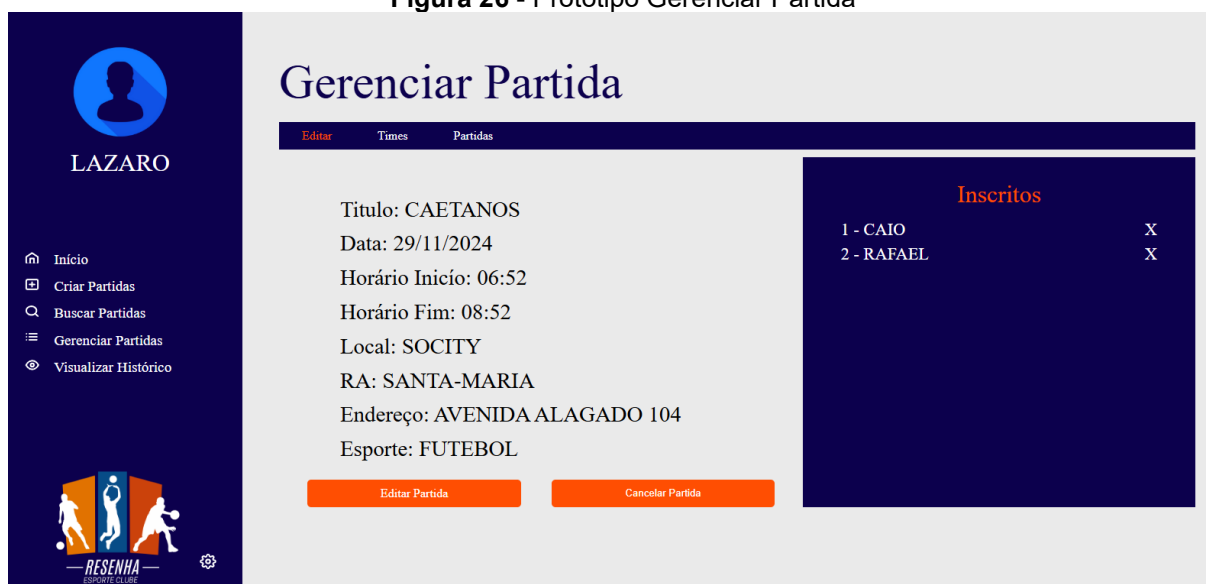
1. A partida pode ter sido cancelada.
2. O sistema pode estar fora do ar.
3. O caso de uso se encerra.

Fonte: Autores, 2024

### 16.2.8.2 Protótipo

A tela a seguir permite a visualização da partida, os jogadores inscritos, cancelar a partida ou editar a mesma.

Figura 26 - Protótipo Gerenciar Partida



Fonte: Autores, 2024

### 16.2.9 Sortear Times

Neste caso de uso onde o ator principal é o jogador, deve possibilitar o sorteio dos times da partida.

#### 16.2.9.1 Detalhamento do Caso de Uso

Quadro 19 - Especificação de Caso de Uso Sortear Times

Nome do Caso de Uso	Sortear Times.
Atores Principais	Jogador.

<b>Atores Secundários</b>	Sistema.
<b>Resumo</b>	Este caso de uso descreve como sortear os times aleatoriamente dentro do sistema.
<b>Pré-Condições</b>	O jogador deve ter executado o caso de uso “Autenticar Jogador”. O jogador deve possuir cadastro ativo no sistema. O jogador deve ter executado o caso de uso “Gerenciar Partida”.
<b>Pós-Condições</b>	Os times serão sorteados.
<b>Fluxo Principal</b>	
<ol style="list-style-type: none"> <li>1. O jogador seleciona a opção “Times” no menu.</li> <li>2. O jogador informa a quantidade de jogadores por time.</li> <li>3. O jogador aciona o botão “Sortear”.</li> <li>4. O sistema sorteia os times.</li> <li>5. O sistema exibe os times sorteados.</li> <li>6. O atleta aciona a opção “Concluir”.</li> <li>7. O sistema salva os times na base de dados.</li> <li>8. O caso de uso se encerra.</li> </ol>	
<b>Fluxo de Exceção</b>	
<ol style="list-style-type: none"> <li>1. A partida pode ter sido cancelada.</li> <li>2. O número de times disponíveis não atende ao mínimo necessário para realizar a partida.</li> <li>3. O sistema pode estar fora do ar.</li> <li>4. O caso de uso se encerra.</li> </ol>	

Fonte: Autores, 2024

### 16.2.9.2 Protótipo

A tela a seguir permite sortear os times aleatoriamente e visualizar os times criados.

Figura 27 - Protótipo Sortear Times



Fonte: Autores, 2024

### 16.2.10 Cronometrar Partida

Neste caso de uso onde o ator principal é o jogador, deve possibilitar cronometrar as partidas de acordo com o esporte.

#### 16.2.10.1 Detalhamento do Caso de Uso

Quadro 20 - Especificação de Caso de Uso Cronometrar Partida

<b>Nome do Caso de Uso</b>	Cronometrar Partidas.
<b>Atores Principais</b>	Jogador.
<b>Atores Secundários</b>	Sistema.
<b>Resumo</b>	Este caso de uso descreve como o jogador pode cronometrar as partidas dentro do sistema.
<b>Pré-Condições</b>	<p>O jogador deve ter executado o caso de uso "Autenticar Jogador".</p> <p>O jogador deve possuir cadastro ativo no sistema.</p> <p>O jogador deve ter executado o caso de uso "Gerenciar Partida".</p> <p>O jogador deve ter executado o caso de uso "Sortear Times".</p>
<b>Pós-Condições</b>	N/A.
<b>Fluxo Principal</b>	

<ol style="list-style-type: none"><li>1. O jogador seleciona a opção “Partidas” no menu.</li><li>2. O sistema gera um <i>timer</i> com as Opções “Pausar”, “Iniciar” e “Reiniciar”, respectivamente.</li><li>3. O jogador aciona as opções.</li><li>4. O sistema, de acordo com a opção selecionada, pode iniciar, pausar ou reiniciar o timer.</li><li>5. O caso de uso se encerra.</li></ol>
<b>Fluxo de Exceção – Cancelamento da partida</b>
<ol style="list-style-type: none"><li>1. A partida pode ter sido cancelada</li><li>2. O sistema pode estar fora do ar.</li><li>3. O caso de uso se encerra.</li></ol>

Fonte: Autores, 2024

16.2.10.2 Protótipo

A tela a seguir permite cronometrar a partida.

Figura 28 - Protótipo Cronometrar Partida



Fonte: Autores, 2024

16.2.11 Gerar Estatísticas da Partida

Neste caso de uso onde o ator principal é o jogador, deve possibilitar gerar as estatísticas da partida.

### 16.2.11.1 Detalhamento do Caso de Uso

**Quadro 21 - Especificação de Caso de Uso Gerar Estatísticas da Partida**

<b>Nome do Caso de Uso</b>	Gerar Estatísticas da Partida.
<b>Atores Principais</b>	Jogador.
<b>Atores Secundários</b>	Sistema.
<b>Resumo</b>	Este caso de uso descreve como gerar as estatísticas de uma partida dentro do sistema.
<b>Pré-Condições</b>	O jogador deve ter executado o caso de uso “Autenticar Jogador”. O jogador deve possuir cadastro ativo no sistema. O jogador deve ter executado o caso de uso “Gerenciar Partida”.
<b>Pós-Condições</b>	As estatísticas das partidas serão geradas para os jogadores participantes.
<b>Fluxo Principal</b>	
<ol style="list-style-type: none"> <li>1. O jogador seleciona a opção “Partidas” no menu.</li> <li>2. O jogador marca o placar da partida.</li> <li>3. O jogador aciona o botão “Concluir partida”.</li> <li>4. O sistema salva o resultado da partida na base de dados.</li> <li>5. O sistema troca os times de acordo com os resultados da última partida.</li> <li>6. O atleta, para encerrar todas as partidas, aciona o botão “Finalizar Todas as Partidas”.</li> <li>7. O sistema exibe as estatísticas da partida para o jogador.</li> <li>8. O caso de uso se encerra.</li> </ol>	
<b>Fluxo de Exceção</b>	
<ol style="list-style-type: none"> <li>1. Os dados das partidas podem não ter sido inseridos, logo não foi possível gerar estatísticas.</li> <li>2. A partida pode não ter dados pois o criador da partida não usou o sistema para gerar estatísticas.</li> <li>3. O sistema pode estar fora do ar.</li> <li>4. O caso de uso se encerra.</li> </ol>	

**Fonte: Autores, 2024**

### 16.2.11.2 Protótipo

A tela a seguir permite gerar as estatísticas da partida de acordo com os placares de cada partida que acontecer.

**Figura 29 - Protótipo Gerar Estatísticas da Partida**



Fonte: Autores, 2024

## 17. ARTEFATOS

Segundo Dantas (2016), um projeto de banco de dados deve apresentar a realidade de forma abstrata, onde as informações possam ser inseridas em uma base de dados.

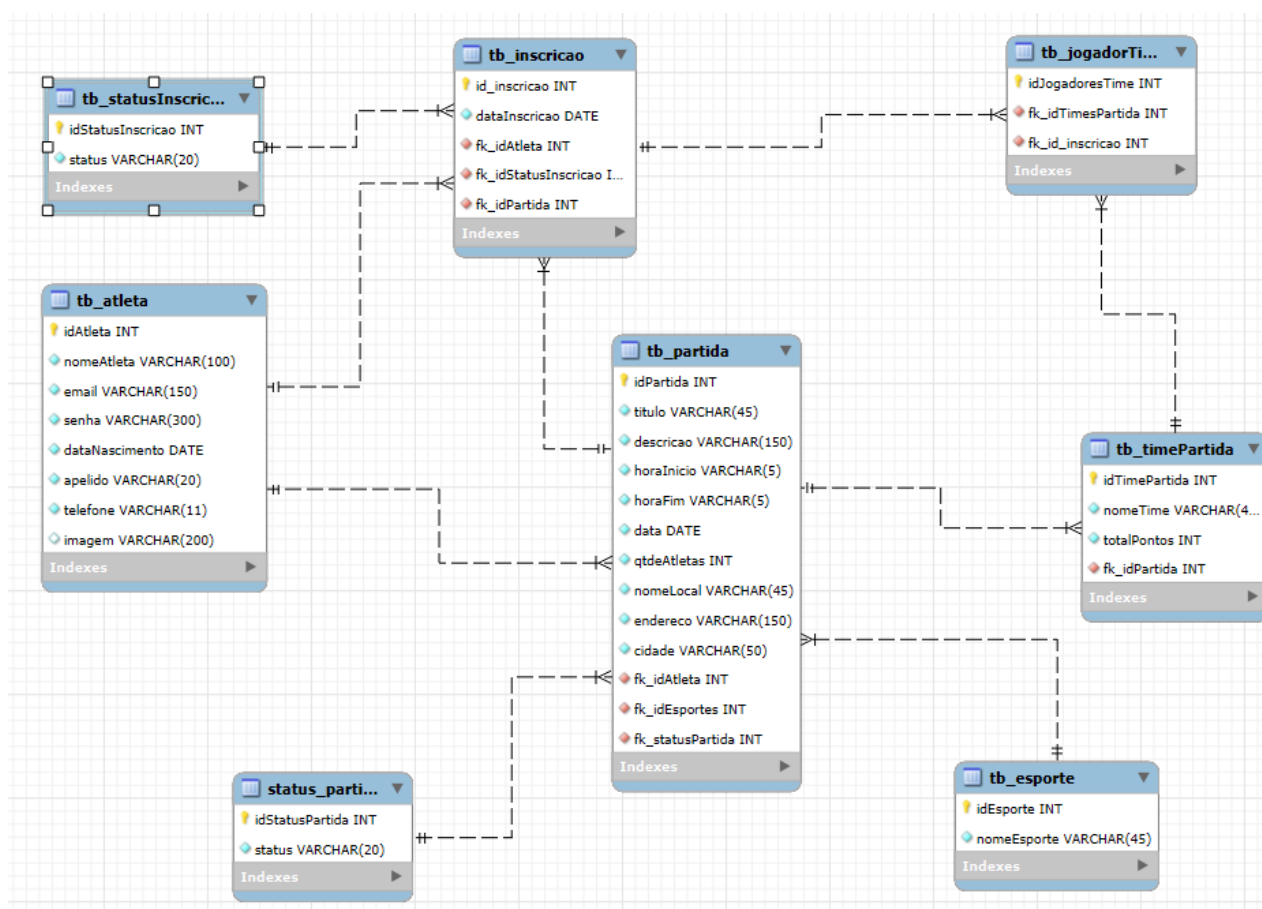
### 17.1. Modelo Entidade Relacionamento (MER)

O Modelo Entidade Relacionamento (MER) é uma ferramenta fundamental para descrever elementos do mundo real de forma abstrata em objetos representados como “entidades”, que podem ser físicas ou conceituais, e pelas conexões entre elas chamadas de “relacionamentos”. Algumas entidades podem ser independentes, existir sem ter um relacionamento com outro objeto, enquanto outras necessitam de um relacionamento com outra entidade para ter sentido no contexto do modelo.

A partir disso, na figura abaixo, está representado o MER do sistema Resenha Esporte Clube.



**Figura 30 – Modelo Entidade Relacionamento**



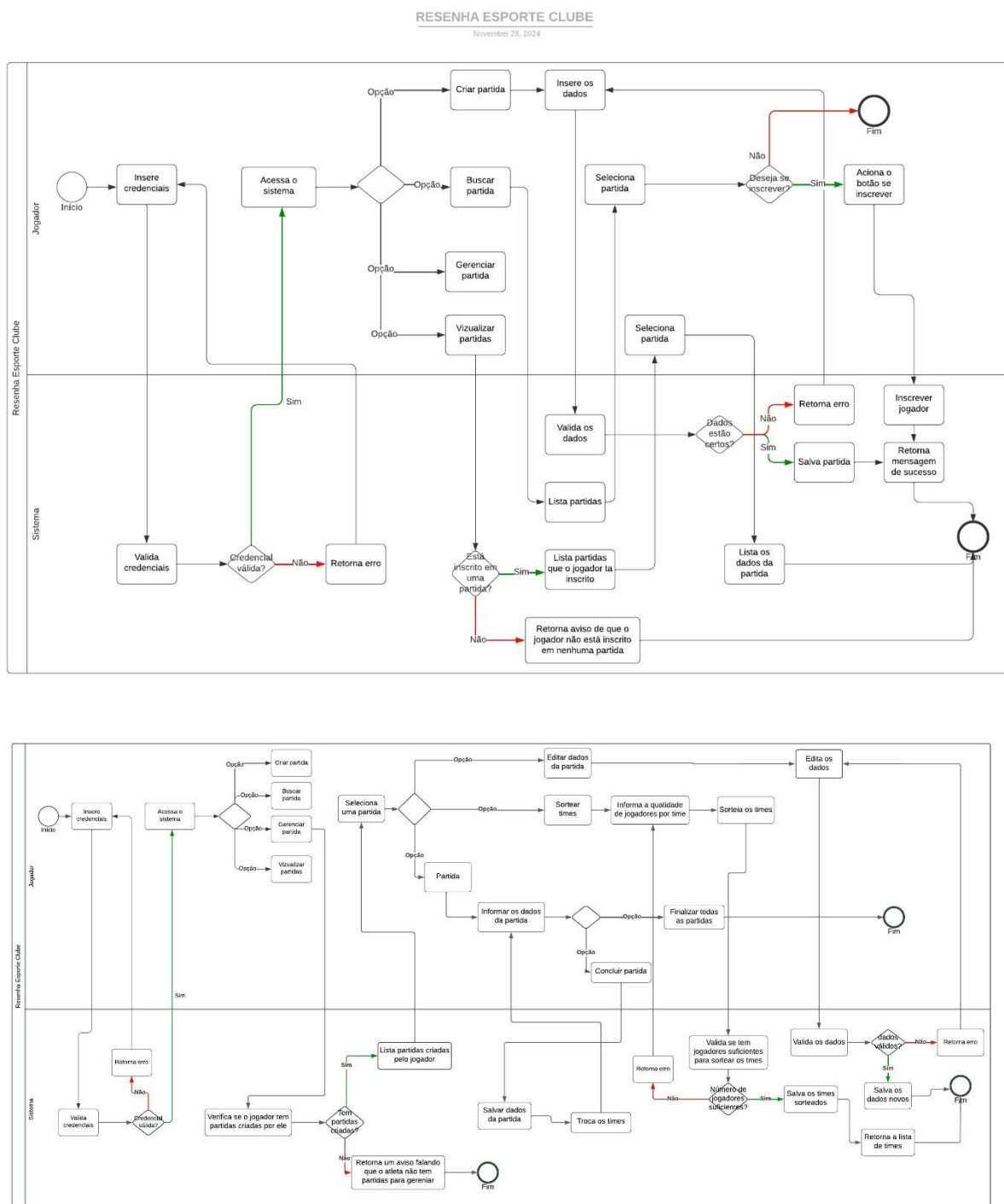
Fonte: Autores, 2024

## 17.2. BPMN

De acordo com Ribeiro (2015), o *Business Process Modeling Notation* (BPMN) desenvolvido em 2004 pela *Business Process Management*, é um padrão de símbolos inseridos em digramas que tem como intuito criar uma linguagem de fácil entendimento para todos os envolvidos na modelagem dos processos de negócio. O objetivo dessa ferramenta/linguagem é aproximar a perspectiva de processos à visão da tecnologia da informação. Os diagramas são compostos pelos seguintes elementos: atividades, eventos, gateways (símbolos de decisões), conectores, agrupamento (Pool) e raia (Lane)".

Abaixo, são representados os diagramas que modelam os processos do sistema Resenha Esporte Clube.

**Figura 31 e 32 – Diagrama BPMN**



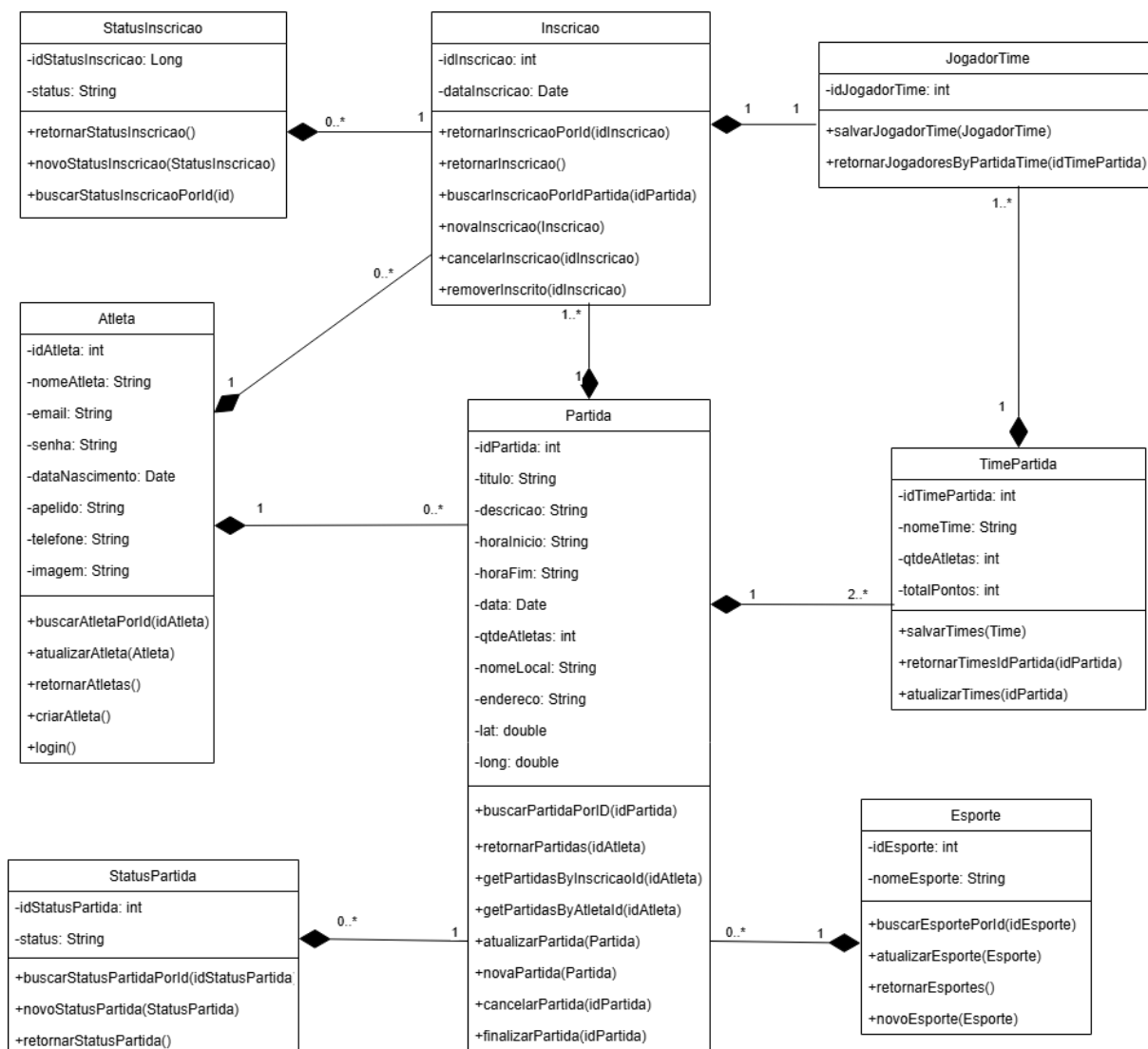
Fonte: Autores, 2024

### 17.3. Diagrama de Classe

Segundo Guedes (2018), o diagrama de classes é uma das ferramentas mais utilizadas e importantes dentro do UML. Seu principal objetivo é permitir uma melhor

visualização das classes que irão compor o sistema, bem como seus métodos e atributos. Em geral, esse diagrama apresenta uma visão estática de como as classes são organizadas dentro do sistema, focando na definição da estrutura lógica entre elas (GUEDES, 2018).

**Figura 33 – Diagrama de Classe**



Fonte: Autores, 2024

## 18. CONSIDERAÇÕES FINAIS

A implementação deste trabalho viabilizou o estudo e a realização do sistema web Resenha Esporte Clube, voltado para a criação e o gerenciamento de partidas esportivas. A aplicação foi desenvolvida com tecnologias atuais do mercado, proporcionando a aplicabilidade dos conhecimentos adquiridos ao longo de toda a trajetória acadêmica na realização deste Trabalho de Conclusão de Curso.

Além disso, por meio das pesquisas de referências teóricas, que fundamentaram a elaboração deste sistema, foi possível realizar uma análise criteriosa e uma avaliação aprofundada dos conceitos e das ferramentas que auxiliaram a construção do Resenha Esporte Clube.

Dessa forma, pode-se compreender que a implementação da ferramenta web é essencial para a organização eficiente de partidas, oferecendo flexibilidade para que os usuários organizem e encontrem jogos de qualquer local.

Os objetivos propostos nesse trabalho foram alcançados, pois o sistema permite a criação de partidas esportivas, permitindo o gerenciamento pelo criador, além da busca e visualização desses eventos pelos usuários da aplicação.

A motivação da execução desse trabalho teve origem na identificação de dificuldades comuns para encontrar e organizar partidas esportivas, levando ao levantamento de requisitos específicos para o sistema. Em seguida, os protótipos do sistema foram desenvolvidos a fim de representar graficamente o seu funcionamento, criando uma base para o auxílio do desenvolvimento. Com essa estruturação definida, deu-se início ao processo de desenvolvimento do sistema, seguindo as modelagens e prototipagens elaboradas.

Em trabalhos futuros, pode-se desenvolver um aplicativo *mobile* que permita ao jogador acessar todas as funcionalidades do sistema sem a necessidade de utilizar o navegador. Além disso, melhorias incluem a implementação de recursos para a criação de partidas públicas e privadas, a possibilidade de reservar quadras, bem como a integração de propagandas de artigos esportivos, disponibilizando aos usuários acesso rápido aos produtos que envolvem esporte.

Outras funcionalidades que podem ser implementadas são: a inserção de novas modalidades esportivas dentro do sistema, o envio de e-mails com atualizações sobre a situação das partidas e a utilização de uma API de geolocalização, permitindo encontrar partidas próximas à localização de cada jogador. Essas melhorias auxiliam para o aprimoramento da experiência do usuário, promovendo cada vez mais a prática esportiva.

## 19. REFERÊNCIAS

ALLEN, J. **Java: The Complete Reference**. 12ª ed. New York: McGraw-Hill, 2020.

ANGULAR. **O que é angular?** Disponível em: <https://angular.dev/overview>. Acesso em: 14 nov. 2024.

APPITO. **É o aplicativo perfeito para que você possa organizar seu futebol, do começo ao fim.** Disponível em PlayStore:<https://play.google.com/store/apps/details?id=com.apitador.app>. Acesso em: 20 set. 2024.

ATLASSIAN. **O que é Scrum.** Disponível em: <https://www.atlassian.com/br/agile/scrum>. Acesso em 24 out. 2024.

BAGLIOTTI, I. R. **Reusabilidade No Desenvolvimento De Um Sistema Web Utilizando O Framework Angular**. Interface Tecnológica - v. 17 n. 1, 2020.

BAUER, C.; KING, G.; GREGORY, G. **Java persistence with hibernate**. 2. ed. New York, NY: Manning Publications, 2015.

CANTELON, M. et al. **Node.js in Action, Second Edition**. 2. ed. New York, NY: Manning Publications, 2014.

CHEGA+. **O Chega+ é o melhor aplicativo do Mundo para organizar grupos de futebol.** PlayStore: <https://play.google.com/store/apps/details?id=app.chegamais.com.chegamais>. Acesso em: 20 set. 2024.

CODIFICADO, C. **O que é MVC?** Disponível em: <https://cafecodificado.com.br/o-que-e-mvc/>. Acesso em: 26 out. 2024.

CORTÉS, M. I. **Fundamentos de Engenharia de Software**. 2013. Disponível em: <https://www.uece.br/cct/wp-content/uploads/sites/28/2021/07/Fundamentos-de-Engenharia-de-Software.pdf>. Acesso em: 16 out. 2024.

COSTA, M. P. S. **Análise de computação em nuvem**. 2023. Disponível em: [https://monografias.ufop.br/bitstream/35400000/5382/6/MONOGRAFIA\\_AnáliseComputaçãoNuvem.pdf](https://monografias.ufop.br/bitstream/35400000/5382/6/MONOGRAFIA_AnáliseComputaçãoNuvem.pdf). Acesso em: 17 out. 2024.

CRUZ, F. **Scrum e PMBoK Unidos no Ferenciamento de Projetos**. BRASPORT, 2013.

CRUZ, T. **Sistemas de informações gerenciais: tecnologia da informação e a empresa do século XXI**. 2. ed. São Paulo: Atlas, 2000.

CUNHA, F. **Sistema Web: o que é e como funciona**. Disponível em: <https://www.mestresdawe.com.br/tecnologias/sistema-web-o-que-e-e-como-funciona>. Acesso em: 17 out. 2024.

CUNHA, F. **Requisitos funcionais e não funcionais: o que são?** Disponível em: <https://www.mestresdawe.com.br/tecnologias/requisitos-funcionais-e-nao-funcionais-o-que-sao>. Acesso em: 17 out. 2024.

DANTAS, C. M.; CORDULA, F. R.; ARAUJO, W. J. **Análise da representação da informação em modelos entidade relacionamento com base em metadados**. Archeion Online, João Pessoa, v.4, n.1, p.40-63, jan./jun. 2016. Disponível em: [https://www.researchgate.net/profile/Flavio-Cordula/publication/331787461\\_ANALISE\\_DA\\_REPRESENTACAO\\_DA\\_INFORMACAO\\_EM\\_MODELOS\\_ENTIDADE\\_RELACIONAMENTO\\_COM\\_BASE\\_EM\\_METADADOS/links/5c8bb4a6a6fdcc381755bbb5/ANALISE-DA-REPRESENTACAO-DA-INFORMACAO-EM-MODELOS-ENTIDADE-RELACIONAMENTO-COM-BASE-EM-METADADOS.pdf](https://www.researchgate.net/profile/Flavio-Cordula/publication/331787461_ANALISE_DA_REPRESENTACAO_DA_INFORMACAO_EM_MODELOS_ENTIDADE_RELACIONAMENTO_COM_BASE_EM_METADADOS/links/5c8bb4a6a6fdcc381755bbb5/ANALISE-DA-REPRESENTACAO-DA-INFORMACAO-EM-MODELOS-ENTIDADE-RELACIONAMENTO-COM-BASE-EM-METADADOS.pdf). Acesso em: 10 out. 2024.

DE LEMOS, M. F. et al. **Aplicabilidade da Arquitetura MVC - Revista Eletrônica Científica de Ciência da Computação**, v. 8, n. 1, 2013. Disponível em: <https://revistas.unifenas.br/index.php/RE3C/article/view/54>. Acesso em: 17 out. 2024.

DEITEL, P.; DEITEL, H. **Java: Como programar**. 10<sup>a</sup> ed. São Paulo: Pearson Education, 2017.

FERREIRA, B. A. **Gestão de Riscos em Projetos: Uma Análise Comparativa da Norma ISO 31000 e o Guia PMBOK®**. Disponível em: <https://periodicos.uninove.br/gep/article/view/9582>. Acesso em: 20 out. 2024.

FINTTA. **Convoque os amigos para disputar as melhores partidas. Organize equipes, agende jogos, grave os lances com o fintta**. Disponível em: PlayStore: <https://play.google.com/store/apps/details?id=com.fintta.app> App Store: <https://apps.apple.com/us/app/fintta-play-soccer-today/id934328306>. Acesso em: 20 set. 2024.

FLINCO, L. O **que é Trello?** Disponível em: <https://blog.planetsweb.com.br/26/06/2020/lais-flinco/o-que-e-trello/>. Acesso em: 26 out. 2024.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2007.

GITHUB. **GitHub Features Overview**. 2023. Disponível em: <https://github.com/about>. Acesso em: 17 out. 2024.

GUEDES, G. T. A. **UML 2 - Uma Abordagem Prática**. [s.l.] Novatec Editora, 2018.

HANASHIRO, A. **VS Code: o que é e por que você deve usar?**, 2021. Disponível em: <https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar>. Acesso em: 17 out. 2024.



HORSTMANN, C. **Core Java Volume I–Fundamentals**. 11<sup>a</sup> ed. Upper Saddle River: Prentice Hall, 2019.

IBGE. **Pesquisa Nacional por Amostra de Domicílios Contínua: Prática de Esportes e Atividade Física**. 2020. Disponível em: <https://www.ibge.gov.br>. Acesso em: 17 out. 2024.

IBGE. **40,3% dos adultos são considerados sedentários no país. Agência Brasil, Rio de Janeiro**. Disponível em: <https://agenciabrasil.ebc.com.br/saude/noticia/2020-11/ibge-403-dos-adultos-sao-considerados-sedentarios-no-brasil>. Acesso em: 23 out. 2024.

JETBRAINS. **IntelliJ IDEA Documentation**. Disponível em: <https://www.jetbrains.com/idea/documentation>. Acesso em: 25 out. 2024.

KULKARNI, V. **Postman API Testing**. 1<sup>a</sup> ed. Birmingham: Packt Publishing, 2020.

LACOMBE, F. J. M.; HEILBORN, G. L. J. **Administração, princípios e tendências**. 1. ed. São Paulo: Saraiva, 2003.

LAUDON, K. C.; LAUDON, J. P. **Sistemas de informação gerenciais: administrando a empresa digital**. 5<sup>a</sup> ed. São Paulo: Pearson Prentice Hall, 2009.

MARZALL, L. F.; DOS SANTOS, L. A.; GODOY, L. P. **Inovação no projeto de produto como fator para redução de custos logísticos e de produção. Revista Produção Online**, [S. l.], v. 16, n. 1, p. 342–365, 2016. DOI: 10.14488/1676-1901.v16i1.2168. Disponível em: <https://www.producaoonline.org.br/rpo/article/view/2168>. Acesso em: 12 nov. 2024.

MINISTÉRIO DA SAÚDE. **Vigitel Brasil 2006-2023: vigilância de fatores de risco e proteção para doenças crônicas por inquérito telefônico**. 1<sup>a</sup> ed. Brasília, DF: Ministério da Saúde, 2024. Disponível em: [https://bvsms.saude.gov.br/bvs/publicacoes/vigitel\\_2006\\_2023\\_pratica\\_atividade\\_fisica.pdf](https://bvsms.saude.gov.br/bvs/publicacoes/vigitel_2006_2023_pratica_atividade_fisica.pdf). Acesso em: 23 nov. 2024.

O'BRIEN, J. A. **Sistemas de informação e as decisões gerenciais na era da Internet**. 2ª ed. São Paulo: Saraiva, 2004.

OLIVEIRA, J. F. **Sistema de Informação: um enfoque gerencial inserido no contexto empresarial e tecnológico**. 1. ed. São Paulo: Érica, 1993.

PAULA, L. **Computação em nuvem: os desafios das empresas ao migrar para a nuvem**. 2021. Disponível em: <https://revista.fatectq.edu.br/interfacetecnologica/article/view/1304/689>. Acesso em: 17 out. 2024.

PEDROSA, P. H. C. **Computação em Nuvem**. 2011. Disponível em: <https://www.ic.unicamp.br/~ducatte/mo401/1s2011/T2/Artigos/G04-095352-120531-t2.pdf>. Acesso em: 17 out. 2024.

PROJECT MANAGEMENT INSTITUTE. **Um guia do conhecimento em gerenciamento de projetos (guia PMBOK)**. 5ª ed. 2013. Disponível em: [https://wiki.tce.go.gov.br/lib/exe/fetch.php/acervo\\_digital:pmbok5.pdf](https://wiki.tce.go.gov.br/lib/exe/fetch.php/acervo_digital:pmbok5.pdf). Acesso em: 16 out. 2024.

RACHID, C. L.; STOPA, G. R. **Scrum: Metodologia Ágil Como Ferramenta De Gerenciamento De Projetos**. CES Revista, v. 33, n. 1, p. 302–323, 2019. Acesso em: 16 out. 2024.

RIBEIRO, T. O. et al. **Benefícios do BPMN na modelagem dos processos: um estudo exploratório**. Engenharia de Produção, Bauru, SP, Brasil, nov. 2015. Disponível em: [https://www.researchgate.net/profile/Tiago-Ribeiro-23/publication/318792227\\_BENEFICIOS\\_DO\\_BPMN\\_NA\\_MODELAGEM\\_DOS\\_PROCESSOS\\_UM\\_ESTUDO\\_EXPLORATORIO/links/597f2a1daca272d5681875d3/BENEFICIOS-DO-BPMN-NA-MODELAGEM-DOS-PROCESSOS-UM-ESTUDO-EXPLORATORIO.pdf](https://www.researchgate.net/profile/Tiago-Ribeiro-23/publication/318792227_BENEFICIOS_DO_BPMN_NA_MODELAGEM_DOS_PROCESSOS_UM_ESTUDO_EXPLORATORIO/links/597f2a1daca272d5681875d3/BENEFICIOS-DO-BPMN-NA-MODELAGEM-DOS-PROCESSOS-UM-ESTUDO-EXPLORATORIO.pdf). Acesso em: 13 out. 2024.

SAMULSKI, D. M; NOCE, F. **A importância da atividade física para a saúde e qualidade de vida: um estudo entre professores, alunos e funcionários UFMG.** Rev. Bras. Atividade Física e Saúde. V. 5, n.1, 2000.

SHARMA, J.; SARIN, A. **Getting Started with Spring Framework: Covers Spring 5.** 4. ed. Birmingham: CreateSpace Independent Publishing Platform, 2020.

SIERRA, K.; BATES, B. **Use a cabeça! Java.** 2ª ed. São Paulo: Alta Books, 2014.

SOMMERVILLE, I. **Engenharia de Software.** 10ª ed. São Paulo: Pearson Prentice Hall, 2019.

SONATYPE. **Maven: The Complete Reference.** 4ª ed., 2020.

TILKOV, S.; VINOSKI, S. **Node.js: Using JavaScript to Build High-Performance Network Programs.** IEEE Internet Computing, v. 14, n. 6, p. 80-83, 2010.

WALLS, C. **Spring in action.** 6. ed. New York, NY: Manning Publications, 2022.