

**CENTRO UNIVERSITÁRIO PARA O DESENVOLVIMENTO DO ALTO VALE DO
ITAJAÍ - UNIDAVI**

GLAYCON FRANCISCO CHIROLI

**MATCHSTART: PROTÓTIPO DE SOFTWARE PARA AGENDAMENTO DE
PARTIDA PARA EQUIPES CASUAIS**

**RIO DO SUL
2023**

**CENTRO UNIVERSITÁRIO PARA O DESENVOLVIMENTO DO ALTO VALE DO
ITAJAÍ - UNIDAVI**

GLAYCON FRANCISCO CHIROLLI

**MATCHSTART: PROTÓTIPO DE SOFTWARE PARA AGENDAMENTO DE
PARTIDA PARA EQUIPES CASUAIS**

Trabalho de Conclusão de Curso a ser apresentado ao curso de Sistemas da Informação, da Área das Ciências Naturais, da Computação e das Engenharias, do Centro Universitário para o Desenvolvimento do Alto Vale do Itajaí, como condição parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Orientador: Esp. Cleber Nardelli

**RIO DO SUL
2023**

**CENTRO UNIVERSITÁRIO PARA O DESENVOLVIMENTO DO ALTO VALE DO
ITAJAÍ - UNIDAVI**

GLAYCON FRANCISCO CHIROLLI

**PROTÓTIPO DE SOFTWARE PARA AGENDAMENTO DE PARTIDA PARA
EQUIPES CASUAIS**

Trabalho de Conclusão de Curso a ser apresentado ao curso de Sistemas da Informação, da Área das Ciências Naturais, da Computação e das Engenharias, do Centro Universitário para o Desenvolvimento do Alto Vale do Itajaí- UNIDAVI, a ser apreciado pela Banca Examinadora, formada por:

Professor Orientador: Esp. Cleber Nardelli

Banca Examinadora:

Prof. Fernando Bastos

Prof. Marciel de Liz Santos

Rio do Sul, 28 de novembro de 2023.

Aos meus pais por sempre me apoiar.

AGRADECIMENTOS

Ao meu pai e minha mãe que sempre me apoiaram com tudo e nunca me deixaram desistir de continuar estudando e aos amigos que sempre estiveram do meu lado nos bons e maus momentos.

Aos professores da Unidavi, por compartilharem seus conhecimentos pessoais e profissionais referentes ao curso, principalmente ao professor e orientador deste trabalho, Cleber Nardelli, que me auxiliou a dar continuidade e concluir este projeto.

E por último, meu falecido Nono Francisco, que esteve presente em boa parte dos meus anos na minha vida acadêmica e foi o meu melhor conselheiro.

RESUMO

Ao longo dos séculos, os esportes têm desempenhado um papel fundamental na vida humana, e com a evolução da tecnologia, estando presente no cotidiano das pessoas, que podem utilizar os seus dispositivos para inúmeras finalidades, é comum que diversos segmentos da sociedade a utilizem de algum modo, como aproximar pessoas para prática do esporte. Este trabalho tem como objetivo o desenvolvimento do protótipo de um software que auxilie no agendamento de partidas casuais, permitindo o encontro de equipes que queiram disputar uma partida de algum esporte coletivo amador. Para isto foi empregado o método de pesquisa aplicada e descritiva, também foi realizado a revisão da literatura referente as tecnologias e metodologias aplicadas no desenvolvimento do protótipo. O protótipo foi desenvolvido utilizando-se de tecnologias atuais baseadas na linguagem Javascript, o desenvolvimento do front-end foi realizado em ReactJs com a utilização de bibliotecas para auxiliar na padronização. A API back-end foi desenvolvida em NodeJs, utilizando do framework Express. No capítulo de desenvolvimento é detalhado os requisitos, regras de negócio, análises, funcionalidades do sistema e a implementação. O protótipo permite que o usuário se cadastre e crie suas equipes na ferramenta, desta forma possibilitando que o mesmo encontre outras equipes para realizar uma partida casual.

Palavras-Chave: Software, Esporte, Dispositivos.

ABSTRACT

Over the centuries, sports have played a fundamental role in human life, and with the evolution of technology, being present in people's daily lives, who can use their devices for countless purposes, it is common for different segments of society to use them in some way, how to bring people together to practice the sport. This work aims to develop a prototype of software that helps in scheduling casual matches, allowing the meeting of teams that want to compete in an amateur team sport match. For this purpose, the applied and descriptive research method was used, and a literature review was also carried out regarding the technologies and methodologies applied in the development of the prototype. The prototype was developed using current technologies based on the Javascript language, the front-end development was carried out in ReactJs using libraries to assist in standardization. The back-end API was developed in NodeJs, using the Express framework. The development chapter details the requirements, business rules, analyses, system functionalities and implementation. The prototype allows the user to register and create their teams in the tool, thus enabling them to find other teams to play a casual game.

Keywords: Software, Sport, Devices.

LISTA DE FIGURA

Figura 1 – Processo de Software.....	18
Figura 2 – Desenvolvimento seguro de software	21
Figura 3 – Fluxograma do Processo de Desenvolvimento do Protótipo.....	29
Figura 4 - WhatsApp.....	31
Figura 5 – BuscaQuadras.....	32
Figura 6 - Fluxograma do processo de cadastro de usuário	37
Figura 7 - Fluxograma para buscar por equipes disponíveis	37
Figura 8 - Fluxograma para visualizar convites de partidas	38
Figura 9 - Diagrama de entidade e relacionamento.....	38
Figura 10 - Criação de funções com ReactJs	39
Figura 11 - Organização de Rotas da Interface.....	40
Figura 12 - Utilização do Material-UI	40
Figura 13 - Utilização do Axios.....	41
Figura 14 - Rotas com Express.....	41
Figura 15 - Tela de Login Mobile.....	42
Figura 16 - Tela de login.....	43
Figura 17 - Tela de cadastro	44
Figura 18 - Menu de Interações	45
Figura 19 - Menu de Interações Input de Equipe	45
Figura 20 - Menu de Interação Visualização Mobile.....	46
Figura 21 - Interface de Início Partidas Em Casa.....	47
Figura 22 - Interface de Início Partidas Fora de Casa.....	47
Figura 23 - Interface de Início Contato e Observação.....	48
Figura 24 - Interface de Minhas Partidas Criar convite de partida	49
Figura 25 - Interface de Minhas Partidas Convites Pendentes.....	49

Figura 26 - Interface Convites de Partidas	50
Figura 27 - Interface Convites de Partidas Alerta.....	50
Figura 28 - Interface de Notificações.....	51
Figura 29 - Interface de Locais de Atuação.....	52
Figura 30 - Interface Criar/Alterar Equipe	53
Figura 31 - Interface Criar/Alterar Equipe 2.....	53
Figura 32 - Interface Criar/Alterar Equipe Alerta	54
Figura 33 - Interface Criar/Alterar Equipe Edição.....	54
Figura 34 - Interface Alterar Usuário	55

LISTA DE QUADROS

Quadro 1 - Gestão e garantia da qualidade do software	20
Quadro 2 - Comparação entre protótipo x estado da arte	34
Quadro 3 - Regras de negócio	34
Quadro 4 - Requisitos funcionais	35
Quadro 5 - Requisitos não funcionais	36

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JS	JavaScript
PWA	<i>Progressive Web App</i>
REST	<i>Representational State Transfer</i>
RF	Requisito Funcional
RN	Regra de Negócio
SGDB	Sistema Gerenciador de Banco de Dados
UI	<i>User Interface</i>
URL	<i>Uniform Resource Locator</i>
WEB	<i>World Wide Web</i>

SUMÁRIO

1. INTRODUÇÃO	14
1.1 PROBLEMA DE PESQUISA	15
1.2 OBJETIVOS	15
1.2.1 Geral	15
1.2.2 Específicos	15
1.3 JUSTIFICATIVA	15
2. REFERENCIAL TEÓRICO	17
2.1 ENGENHARIA DE SOFTWARE.....	17
2.1.1 Processo	17
2.1.2 Requisitos	18
2.1.3 Qualidade e Segurança	19
2.1.4 Repositório Git	21
2.1.5 Visual Studio Code	21
2.1.6 Unified Modeling Language	21
2.2 LINGUAGENS E LÓGICA DE PROGRAMAÇÃO.....	22
2.2.1 Programação Orientada a Objetos	22
2.3 BANCO DE DADOS	22
2.3.1 Modelo de Dados Relacional	23
2.3.2 PostgreSQL	24
2.3.3 SQL	24
2.4 HTML.....	25
2.4.1 HTML5	25
2.5 CSS	25
2.6 JAVASCRIPT	26
2.7 REACT	26

2.8 NODE.JS	26
2.8.1 NPM	27
2.8.2 EXPRESS	27
2.9 LEI GERAL DE PROTEÇÃO DE DADOS.....	27
3. METODOLOGIA DA PESQUISA	29
3.1 ESTADO DA ARTE	30
3.1.1 WHATSAPP	30
3.1.2 BUSCAQUADRAS.....	31
4. MATCHSTART: PROTÓTIPO DE SOFTWARE PARA AGENDAMENTO DE PARTIDA PARA EQUIPES CASUAIS.....	33
4.1 ANÁLISE.....	33
4.1.1 VISÃO GERAL.....	33
4.1.2 COMPARAÇÃO DO PROTÓTIPO COM O ESTADO DA ARTE.....	34
4.2 REGRAS DE NEGÓCIO	34
4.3 REQUISITOS	35
4.4 DIAGRAMAS.....	36
4.5 IMPLEMENTAÇÃO	39
4.6 UTILIZAÇÃO E FUNCIONAMENTO.....	42
4.6.1 Tela de Login do Usuário	42
4.6.2 Tela de Cadastro de Usuário	43
4.6.3 Menu de Interações.....	44
4.6.4 Interface de Início	46
4.6.5 Interface de Minhas Partidas	48
4.6.6 Interface dos Convites de Partidas.....	50
4.6.7 Interface das Notificações.....	51
4.6.8 Interface dos Locais de Atuação	51
4.6.9 Interface para Criar/Alterar Equipes	52

4.6.10 Interface para Alterar Usuário.....	55
5. CONCLUSÃO	56
5.1 RECOMENDAÇÕES DE TRABALHOS FUTUROS.....	57

1. INTRODUÇÃO

Ao longo dos séculos, os esportes têm desempenhado um papel fundamental na vida humana, e a sua importância só tem crescido com o passar do tempo. Batista (2023) cita que é possível encontrarmos diversas versões que podem explicar a origem do esporte até evoluir para o que conhecemos hoje como práticas esportivas.

Em pesquisa realizada pelo Ministério do Esporte (2015), o esporte é praticado por cerca de 25,6% da população brasileira com idades entre 14 e 75 anos, além de 28,5%, praticarem alguma atividade física.

Dados do instituto Visão (2015) mostram que o futebol, em geral, é o esporte mais praticado em todas as regiões do Brasil, com índices maiores no Sul (67,7% dos praticantes). Em seguida temos o voleibol, onde nas regiões Norte e Centro-Oeste estão os maiores índices (15,20% dos praticantes).

A tecnologia também é algo que vem evoluindo e muito na vida dos seres humanos, estando presente no cotidiano das pessoas, especialmente por conta da vasta opção de conectividade disponível, que podem utilizar de seus próprios dispositivos para inúmeras finalidades, sejam eles móveis ou não.

De acordo com a Cable News Network Brasil (2022), o Brasil já possui mais *Smartphones* do que habitantes, a venda de aparelhos móveis em 2021 teve um crescimento de 27%, com mais de 14 milhões de unidades comercializadas.

De acordo com esses dados, observa-se que a tecnologia, em especial de comunicação móvel, é quase indispensável na vida humana e por conta dessa grande amplitude, é comum que os mais diversos segmentos da sociedade a utilizem de algum modo, bem como auxiliarmos na prática do esporte como um todo.

O objetivo deste trabalho é facilitar o agendamento de partidas casuais, auxiliando no encontro de equipes que queiram disputar uma partida de algum esporte coletivo amador, como por exemplo, no futsal, e para isto foi decidido desenvolver um protótipo de software.

Além do capítulo introdutório, trabalho apresenta na segunda parte uma revisão literária onde será comentado sobre as tecnologias utilizadas no desenvolvimento do protótipo. No terceiro capítulo, está descrito a metodologia de pesquisa. O quarto capítulo é referente ao desenvolvimento e análise do projeto. E por fim o último capítulo lista as considerações finais.

1.1 PROBLEMA DE PESQUISA

Como facilitar a pesquisa por outras equipes praticantes de esporte a fim de realizar uma partida casual utilizando dispositivos pessoais conectados na internet?

1.2 OBJETIVOS

Nesta sessão serão detalhados os objetivos do trabalho.

1.2.1 Geral

Desenvolver protótipo de aplicação web para equipes amadoras de esportes coletivos marcarem partidas casuais.

1.2.2 Específicos

- Definir as regras de negócio.
- Levantar requisitos para o desenvolvimento.
- Selecionar as ferramentas para implementação.
- Implementar o back-end.
- Desenvolver o front-end responsivo.
- Realizar pesquisa do estado da arte.

1.3 JUSTIFICATIVA

A prática de esportes é uma tendência no Brasil e está aumentando cada vez mais, e em muitos desses tipos de entretenimento, como o futebol e vôlei, é necessário que haja a disputa entre duas equipes, e encontrar um segundo time para realizar uma partida é sempre uma dificuldade, principalmente para quem está iniciando e não conhece outras equipes.

Desta forma, fazer com que as equipes amadoras (principalmente as que estão iniciando) consigam marcar partidas com outras, facilitará a vida para os dois lados, tanto para as que estão por mais tempo neste cenário e querem encontrar novos times

para disputar uma partida, quanto para equipes que estão iniciando e não conhecem quem disputa o esporte na região.

No mercado já existem ferramentas que possibilitam a comunicação entre equipes, como por exemplo o “WhatsApp”, mas nestes aplicativos é necessário que os usuários já se conheçam e possuam o contato um do outro, diferente do protótipo deste trabalho, que resolverá este problema possibilitando aos usuários encontrarem outras equipes na região.

Neste contexto o presente trabalho visa desenvolver um protótipo de aplicativo que atenderá estas necessidades, possibilitando que os usuários procurem por outras equipes que queiram disputar uma partida casual e facilitando a organização dos horários e locais destas partidas.

2. REFERENCIAL TEÓRICO

Nesse capítulo serão abordados assuntos referentes as tecnologias, ferramentas de desenvolvimento e métodos utilizados no processo de desenvolvimento da aplicação.

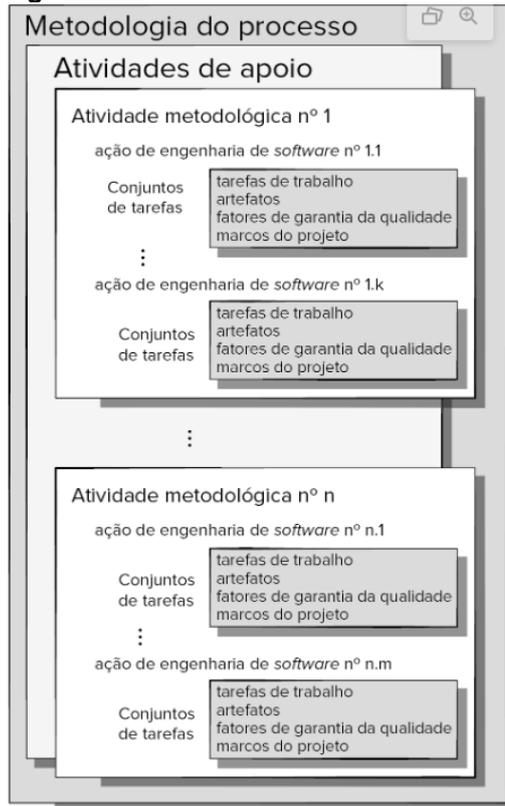
2.1 ENGENHARIA DE SOFTWARE

Pressman e Maxim (2021), dizem que um “Software” são instruções que, quando executadas, fornecem características, funções e desempenhos desejados, como estrutura de dados que permitem aos programas manipularem as informações adequadamente.

Também para Pressman e Maxim (2021, p.8), “Engenharia de software: A aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software, isto é, a aplicação de engenharia ao software.”.

2.1.1 Processo

De acordo com Pressman e Maxim (2021), no processo de software as atividades metodológicas são compostas por um conjunto de ações de engenharia de software. Cada ação será definida por um grupo de tarefas, que identificam as ações que devem ser completadas, os artefatos de software que serão produzidos, fatores de garantia da qualidade que serão exigidos e os marcos utilizados para indicar o progresso. A Figura 1 ilustra o processo.

Figura 1 – Processo de Software

Fonte: Pressman e Maxim (2021, p. 20).

Para Schach (2010), diferentes empresas utilizam de processos de softwares diferentes, um exemplo seria a questão da documentação. Algumas empresas documentam de forma intensiva seus produtos, já outras dizem que o produto pode ser compreendido automaticamente ao ler seu código-fonte. O autor também afirma que os testes são outra forma de comparar empresas. Certas empresas usam de metade de seus orçamentos com testes e acreditam que somente os usuários do software podem testá-lo completamente. Diferente de companhias que dedicam mais tempo corrigindo erros e problemas que são relatados pelos próprios usuários do que testando o produto.

2.1.2 Requisitos

De acordo com Reinehr (2020), um requisito funcional é uma funcionalidade que os stakeholders utilizam para cumprir objetivos de negócio, representando o que os desenvolvedores deverão implementar para que os usuários consigam realizar suas atividades. Já os requisitos de qualidade são referenciados como o software que

operará sobre circunstâncias determinadas, estes estão relacionados a questões como desempenho, disponibilidade, usabilidade, etc.

Entender os requisitos de um problema está entre as tarefas mais difíceis enfrentadas por um engenheiro de software. Quando você pensa nisso pela primeira vez, entender claramente a engenharia de requisitos não parece assim tão difícil. Afinal, o cliente não sabe o que é necessário? Os usuários não deveriam ter um bom entendimento das características e funções de que precisam? Surpreendentemente, em muitos casos, a resposta a essas perguntas é “não”. E mesmo que os clientes e usuários soubessem explicitar as suas necessidades, estas mudariam ao longo do projeto. (PRESSMAN; MAXIM, 2021, p. 102)

Para Pressman e Maxim (2021) o levantamento de requisitos é um processo que parece simples, mas isto, na verdade é muito difícil. Uma das partes importantes deste processo é entender quais são as metas do negócio, pois elas ajudam a explicar os requisitos para os envolvidos, e podem ser usadas para resolver conflitos. Quando capturadas, as metas estabelecem um mecanismo de atribuição de prioridades e criam um raciocínio lógico para a possível arquitetura do projeto. Levantar requisitos é transferir as ideias dos envolvidos para a equipe de software de forma congruente e sem falhas.

2.1.3 Qualidade e Segurança

De acordo com Vetorazzo (2018), planejar a qualidade significa utilizar de padrões apropriados para o produto e o processo. Isto definirá os atributos que possuem maior significado e de que forma eles serão avaliados. O autor também informa que para se possuir um bom plano de qualidade deve-se apresentar a introdução ao produto, os planos de produto, as descrições de processos, as metas de qualidade e os riscos e gerenciamentos de riscos.

Para Pressman e Maxim (2021), a garantia e qualidade são atividades essenciais em qualquer empresa que terão seus produtos utilizados por terceiros, ela engloba um amplo espectro de preocupações e atividades que se concentram na gestão da qualidade do software. Podem ser sintetizadas da seguinte maneira indicada no quadro 1:

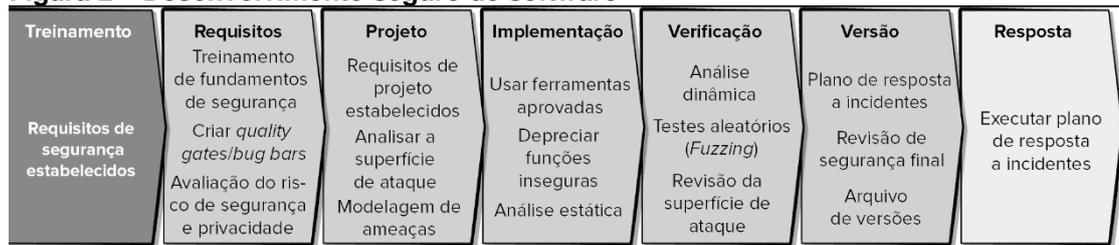
Quadro 1 – Gestão e garantia da qualidade do software

Padrões	Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE), a ISO e outras organizações de padronizações produziram uma ampla gama de padrões para engenharia de software e documentos relacionados. Os padrões podem ser adotados voluntariamente por uma organização de engenharia de software ou impostos pelo cliente ou outros envolvidos.
Revisões e auditorias	As revisões técnicas são uma atividade de controle de qualidade realizada por engenheiros de software para engenheiros de software. Seu intuito é o de revelar erros. Auditorias são um tipo de revisão realizada pelo pessoal de SQA com o intuito de assegurar que as diretrizes de qualidade estejam sendo seguidas no trabalho de engenharia de software.
Testes	São uma função de controle de qualidade com um objetivo principal: encontrar erros.
Coleta e análise de erros/defeitos	A única forma de melhorar é medir o nosso desempenho. Reúne e analisa dados de erros e defeitos para melhor compreender como os erros são introduzidos e quais atividades de engenharia de software são as mais adequadas para sua eliminação.
Gerenciamento de mudanças	São um dos aspectos mais disruptivos de qualquer projeto de software. Se não forem administradas podem gerar confusão.
Administração da segurança	Com o aumento dos crimes cibernéticos e novas regulamentações governamentais referentes à privacidade, toda organização de software deve instituir políticas que protejam os dados em todos os níveis, estabelecer proteção por meio de <i>firewalls</i> e garantir que o software não tenha sido alterado internamente sem autorização.
Proteção	Em razão de o software ser quase sempre um componente fundamental de sistema que envolvem vidas humanas, o impacto de defeitos ocultos pode ser catastrófico.
Gestão de riscos	Embora a análise e a redução de riscos sejam preocupações dos engenheiros de software, o grupo de Garantia de Qualidade de Software (SQA) garante que as atividades de gestão de riscos sejam conduzidas apropriadamente e que planos de contingência relacionados a riscos tenham sido estabelecidos.

Fonte: Elaborado a partir de Pressman e Maxim (2021).

Conforme Pressman e Maxim (2021), segurança de software é mais do que proteger o software operacional usando *firewalls*, senhas fortes e criptografia. Trata-se de uma forma mais segura de desenvolver o software desde o início. Nos dias atuais, existem técnicas disponíveis que ajudam a desenvolver softwares significativamente mais seguros do que seria possível no passado. A engenharia de segurança de software é uma área bastante fértil.

Continuando com Pressman e Maxim (2021), o SDL (ciclo de vida do software do desenvolvimento de segurança) é um processo de segurança de software líder no setor, abrange toda a empresa e é política obrigatória dentro dela desde 2004, o que permitiu que a Microsoft integre segurança e privacidade ao seu software e à sua cultura. O SDL introduz a segurança e privacidade em todas as fases do processo de desenvolvimento, a Figura 2 contém uma representação do modelo de processo de desenvolvimento seguro.

Figura 2 – Desenvolvimento seguro de software

Fonte: Pressman e Maxim (2021, p. 358).

2.1.4 Repositório Git

De acordo com Faria (2017, p. 9), “Pode-se considerar o Git como um minissistema de arquivos, onde todas as alterações no diretório do projeto, e subdiretórios, são controladas. Ele funciona tirando snapshots dos arquivos alterados toda vez que uma operação de *commit* é executada.”

Conforme Faria (2017), o Git é um sistema de versionamento distribuído, que foi criado em 2005 pelos criadores do kernel do Linux, permitindo a criação de *branches* ou ramos, assim quando um desenvolvedor gera uma alteração no código, ele pode optar por criar um ramo no código e depois fundir as alterações com o ramo principal. A grande força do Git é permitir a colaboração entre diferentes desenvolvedores, e para isso ser possível e utilizado de um servidor centralizar que pode ser privado ou contratado como serviço de uma empresa de hospedagem.

2.1.5 Visual Studio Code

Para Fitzgerald, et al. (2021), o VS Code é um editor de código-fonte gratuito que utiliza um sistema de pastas ou espaços de trabalho para interagir com um projeto, e um sistema de documentos que lida com os arquivos. Possui suporte para diversas linguagens de programação como JavaScript e TypeScript, isto por meio de um ecossistema de extensões robusto.

2.1.6 Unified Modeling Language

De acordo com Moraes (2020), com o UML é possível visualizar e especificar detalhes de um software em desenvolvimento por meio de específicas linguagens, permitindo que elementos, relacionamentos e diagramas sejam modelados. Os relacionamentos entre as classes dentro de um contexto de orientação a objetos,

ocorrem por meio de dependências, associações, implementações e até generalizações. A demanda social trouxe diversas versões da UML, que buscaram sempre acompanhar a necessidade dos desenvolvedores de software.

2.2 LINGUAGENS E LÓGICA DE PROGRAMAÇÃO

Para Alves (2014, p. 10), “[...] A lógica nos ensina como utilizar o pensamento da melhor forma possível, em busca da solução de um problema, e está fortemente ligada à matemática e à Filosofia, tendo origem na Grécia antiga.”.

Conforme Sebesta (2018), a linguagem na qual programadores utilizam no início do desenvolvimento de software impõe restrições nos tipos de estruturas de controle, estrutura de dados e abstrações que podem utilizar, limitando a forma com que podem construir seus algoritmos. Para reduzir estas limitações, o programador pode aumentar a diversidade de seus processos mentais de desenvolvimento ao aprender novas construções de linguagens.

2.2.1 Programação Orientada a Objetos

De acordo com Alves (2014), a programação orientada a objeto é uma técnica que surgiu no final dos anos de 1970, quando pesquisadores do Centro de Pesquisas da Xerox, na Califórnia, desenvolveram uma linguagem de programação denominada *SmallTalk*, que usava o conceito de classes e objetos. Essa técnica tem a capacidade de reutilizar códigos já prontos, denominados classes para criamos as partes denominadas objetos, que é a parte vital do programa, economizando espaço e tempo no desenvolvimento de um programa.

2.3 BANCO DE DADOS

Segundo Pichetti, Cortes e Paixão (2021), os bancos de dados surgiram em função da grande necessidade de armazenar dados de forma permanente e organizada. O volume de dados é cada vez maior e com o surgimento dos sistemas de gerenciamento de banco de dados (SGBDs) é possível extrair o máximo de informações que dão auxílio ao processo de tomadas de decisão no formato necessário em pouco minutos. Saber mensurar a capacidade de processamento

exigida por um SGDB determina a escolha da arquitetura mais adequada ao projeto, cada uma contendo suas vantagens e desvantagens.

Para Dias (2021), nos primeiros dias da computação, os dados eram armazenados em fitas, que eram principalmente usadas apenas para gravação. Isso significava que os dados não podiam ser lidos depois de serem armazenados. Essas fitas eram lentas e pesadas, o que levou os cientistas da computação a procurarem uma solução melhor. Um banco de dados é uma maneira de organizar os dados para que as informações desejadas possam ser encontradas e gerenciadas rapidamente. Por exemplo, uma lista de nomes em ordem alfabética ou uma lista de códigos numéricos crescentes, onde cada código representa um item de estoque, são exemplos simples de um banco de dados. Existem diferentes modelos de banco de dados que permitem armazenar informações de várias maneiras.

2.3.1 Modelo de Dados Relacional

De acordo com Alves (2020), o modelo de dados relacional idealizado por Ted Codd atualmente domina o mercado de processamento de dados. Tendo como base a teoria dos conjuntos, utiliza o conceito de relações vindo da Matemática. Esse modelo tem como objetivo representar os dados e os relacionamentos existentes entre eles por meio de uma coleção de tabelas que são compostas por linhas, que são subdivididas em colunas nomeadas, facilitando a interpretação dos dados armazenados.

Para Machado (2020), as colunas de uma relação devem ser atômicas, isto é, indivisíveis em termos de valores e componentes, não existindo colunas do tipo subgrupo. Cada linha de uma tabela representa um objeto, um assunto que é descrito pelos valores de cada uma dessas colunas.

Para Dias (2021, p. 08), “Um banco de dados relacional contém várias tabelas de dados com linhas e colunas que se relacionam entre si, por meio de campos-chave especiais. Esses bancos de dados são mais flexíveis do que as estruturas de arquivo simples e fornecem funcionalidade para leitura, criação, atualização e exclusão de dados. Os bancos de dados relacionais usam *Structured Query Language* (SQL), um aplicativo de usuário padrão que fornece uma interface de programação fácil para interação com o banco de dados.”

Continuando com Machado (2020), em todas as tabelas existentes em um banco de dados relacional, haverá uma coluna ou conjunto de colunas concatenadas, cujos valores são únicos na tabela, nunca se repetindo aquele valor em nenhuma outra linha. Esse conjunto de colunas identifica uma única linha da tabela, denominada de chave primária da tabela.

2.3.2 PostgreSQL

De acordo com Soares (2013), PostgreSQL é um banco de dados relacional que nos últimos anos vem evoluindo de forma consistente, agregando funcionalidades antes disponíveis somente em banco de dados pagos. Esta é uma característica que mais atrai desenvolvedores ao PostgreSQL, pois além de poderoso, versátil, transacionalmente seguro, dispor de *stored procedures* e *triggers*, ele é gratuito.

2.3.3 SQL

De acordo com Machado (2020), *Structured Query Language* (SQL) é de grande utilidade, foi fundamentada no modelo relacional de Codd (1970). Sua primeira versão era denominada SEQUEL (*Structure English Query Language*), mas foi revisado e ampliado, com isso seu nome foi alterado para SQL por razões jurídicas. A linguagem SQL assume um papel muito importante nos Sistemas de Gerenciamento de Banco de dados.

Para Cardoso (2009, p. 14), “A SQL é uma linguagem declarativa, em que, o que importa, no processamento, é o resultado; como e no quê é executado não importa. Essa linguagem é baseada em álgebra e cálculo relacional e possui uma sintaxe bem simples.”.

Continuando com Cardoso (2009), para que possa fornecer tantos recursos o SQL é formado por várias partes, cada uma possuindo seu propósito específico. Nesta composição temos a DDL (*Data Definition Language* – linguagem de definição de dados), a DML (*Data Manipulation Language* – linguagem de manipulação de dados), a DCL (*Data Control Language* – linguagem de controle dos dados), a DTL (*Data Transaction Language* – linguagem de controle dos dados) e por fim a DQL (*Data Query Language* - linguagem de controle dos dados).

2.4 HTML

De acordo com Alves (2021), assim como são necessárias as linguagens de programação para o desenvolvimento de programas e aplicativos, também é necessária uma linguagem para criação de páginas de sites. Essa linguagem é denominada HTML (*HyperText Markup Language* – Linguagem de Marcação de HiperTexto). Esta linguagem não comanda a execução de cálculos ou processamento de dados, ela possui marcadores que são envolvidos pelos sinais “<” e “/>” que indicam o início e o fim da formatação.

Para Miletto e Bertagnolli (2014), o HTML é a base para criar uma página exibida em um navegador WEB. É constituído por um conjunto de TAGs que possibilita exibir o conteúdo e utilizar recursos hipermídia – links, imagens, tabelas e vídeos. Seus recursos de formatação visual são restritos e simples. Uma TAG é uma palavra específica, definida em HTML, envolta por sinais de “menor que” e “maior que”, que indicam o início e fim da marcação.

2.4.1 HTML5

Segundo Alves (2021), o HTML5 consiste em uma nova versão da linguagem HTML que oferece novos recursos ao desenvolvedor web para criação de interfaces mais avançadas, capazes de proporcionar uma melhor experiência ao usuário. Além de praticamente eliminar a necessidade de uso de tecnologia de terceiros, ela oferece ainda, melhores controles de interface. O HTML5 elimina a necessidade do uso de *plug-in Flash* ou *Silverlight* para exibição de vídeo e áudio em uma página web.

2.5 CSS

Miletto e Bertagnolli (2014), definem o CSS (*Cascading Style Sheets*) como diferentes tipos de formatações, como bordas, cores, fundo, elementos textuais estilizados e layouts diferenciados mudando a forma de organização das páginas. O CSS possibilita definir em um único local a formatação que será utilizada por cada TAG. Com isso, apenas um arquivo é alterado, sendo que a mudança é automaticamente propagada a todas as páginas que compõem o site.

2.6 JAVASCRIPT

Flanagan (2014), informa que JavaScript é a linguagem de programação da Web, utilizada pela maioria dos sites modernos, todos os navegadores modernos, computadores de mesa, consoles de jogos, tablets e smartphones, todos estes possuem interpretadores de JavaScript, tornando-a a linguagem de programação mais onipresente da história. A sintaxe de JavaScript é derivada da linguagem *Java*, das funções de primeira classe de *Scheme* e da herança baseada em protótipos de *Self*, sendo uma linguagem de alto nível, dinâmica, interpretada e não tipada. Conveniente para estilos de programação orientados a objetos e funcionais.

Para Oliveira e Zanetti (2020), a linguagem de programação JavaScript é utilizada em páginas web para acrescentar características dinâmicas, possibilitando a interação com usuário de forma mais simples e eficiente, acrescentando inúmeras possibilidades quando aplicado em conjunto com HTML e CSS na construção de sites para internet. Uma das mais importantes características da linguagem JavaScript é que ela não apresenta tipos de dados, isto é, qualquer variável definida é do tipo variante.

2.7 REACT

Lançado em 2013, o React é um framework JavaScript. Foi criado originalmente por engenheiros do Facebook para resolver os desafios envolvidos ao desenvolver interfaces de usuário complexas com conjuntos de dados que mudam ao longo do tempo. O React foi criado para atender a uma necessidade específica relacionada a um conjunto específico de desafios tecnológicos enfrentados pelo Facebook. (GACKENHEIMER, 2015).

O React não tem como objetivo resolver todos os problemas na interface do usuário, mas sim um conjunto específico de problemas, e, em geral um único problema. O React constrói interfaces de usuário em larga escala com dados que mudam ao longo do tempo. (GACKENHEIMER, 2015).

2.8 NODE.JS

De acordo com Oliveira e Zanetti (2021), o desenvolvimento de software para aplicações web é uma área em expansão e em constante mudança, e até mesmo

linguagens tradicionais, como o JavaScript, ganham novas aplicações, como o Node.js, que nada mais é do que um servidor gratuito e de código aberto, que possibilita a criação de aplicações no lado servidor com algumas vantagens em relação aos seus principais concorrentes, no que se refere a leveza, flexibilidade suporte e produtividade. Além disso, o mesmo já está alinhado à modernidade no desenvolvimento de software para a internet.

Para Cardoso (2021), o Node.js permite aos desenvolvedores criar diversos tipos de ferramentas e aplicativos (*back-end*) em JavaScript, é utilizado extensamente de um browser, ele age diretamente no servidor ou também no computador omitindo o ambiente APIs JavaScript do browser.

2.8.1 NPM

O Gerenciador de Pacotes do Node.js (NPM) é uma ferramenta que permite acessar uma ampla variedade de pacotes, com a vantagem de poder utilizá-los várias vezes. O NPM do Node.js possui uma extensa coleção de dependências que podem ser usadas para automatizar grande parte das ferramentas de compilação. Outra vantagem do NPM é sua portabilidade e disponibilidade em várias versões para diferentes sistemas operacionais, incluindo OS X, Microsoft Windows, Linux, NonStop, Solaris, WebOS, FreeBSD e OpenBSD. (CARDOSO, 2021).

2.8.2 EXPRESS

O Express é o framework mais conhecido do Node.js, dispõe de várias soluções para gerenciar as requisições HTTP nas URLs. Permite a integração de *view engines* possibilitando a inserção de dados nos modelos, define os parâmetros rotineiros da aplicação web. Também permite adicionar novos processo de requisição pelo middleware em quaisquer pontos da "fila" de requisições e possibilita que os desenvolvedores criem pacotes específicos de middleware. (CARDOSO, 2021).

2.9 LEI GERAL DE PROTEÇÃO DE DADOS

De acordo com Oliveira (2021), a Lei 13.709, de 14 de agosto de 2018, mas conhecida como Lei Geral de Proteção de Dados Pessoais (LGPD) não encontra fronteiras em segmentos de negócios. Para sua aplicação é simplesmente necessário

estar diante de dados pessoais, sendo um problema comum para quem tem pessoas físicas como cliente ou é uma pessoa jurídica. As sanções administrativas previstas na LGPD variam, mas as mais graves podem resultar na quebra de um negócio.

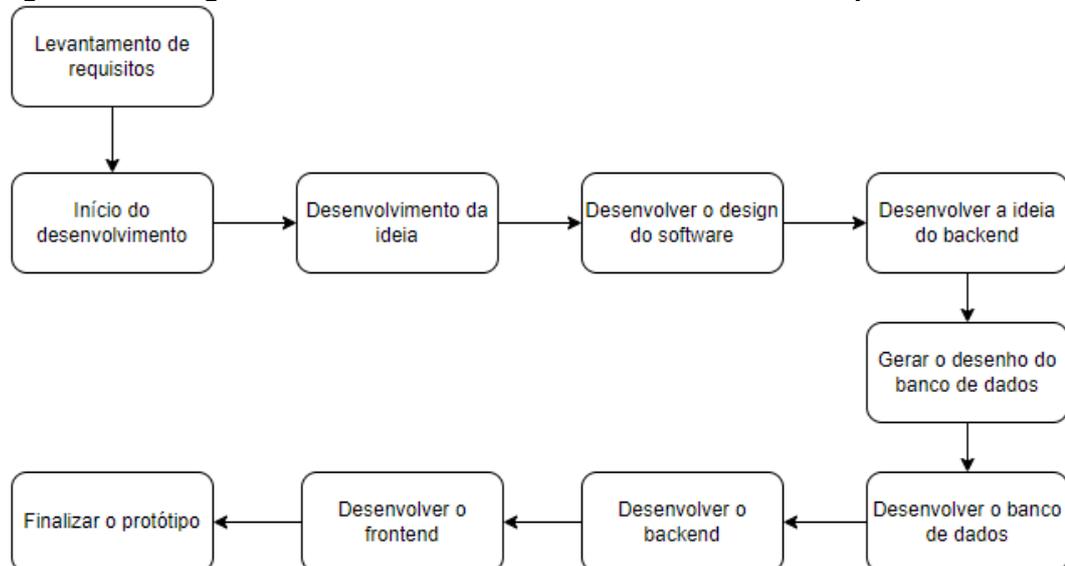
3. METODOLOGIA DA PESQUISA

O presente trabalho de conclusão de curso caracteriza-se como pesquisa aplicada e descritiva, uma vez que tem como objetivo a criação de um protótipo de aplicação web responsiva com rotinas para que equipes casuais possam encontrar outras equipes interessadas em desfrutar de uma partida.

O trabalho tem como foco o seguinte problema: Como facilitar a pesquisa por outras equipes praticantes de esporte a fim de realizar uma partida casual utilizando dispositivos pessoais conectados na internet?

Seguindo o fluxograma apresentado na Figura 3, foram realizadas atividades para entender os princípios e estruturas que precisam ser atingidos para o desenvolvimento do protótipo. Foi definido a criação de duas aplicações, uma que tem como objetivo o *back-end* da aplicação, e a outra o *front-end*, os dois terão seu desenvolvimento com base no uso de Node.JS e JavaScript, comunicando-se por requisições HTTP com transporte de dados no formato JSON.

Figura 3 – Fluxograma do Processo de Desenvolvimento do Protótipo



Fonte: Acervo do Autor (2023).

O *back-end* também conhecido como “servidor”, tem como função gerenciar e processar os dados e a lógica do sistema, atuando nos bastidores fornecendo as funcionalidades e serviços que são acessadas pelo usuário na interface da aplicação. É necessário a conexão com um banco de dados, onde o software recuperará as informações necessárias para a aplicação, além de poder realizar a criação, inclusão, atualização e leitura dos dados.

O *front-end* é a parte da aplicação que trabalha se comunicando diretamente com o usuário, nele são concentrados a apresentação visual e experiência do usuário, tornando a interface amigável e responsiva. Ele inclui o layout, design, cores, tipografia e elementos visuais que compõe a aparência da aplicação. São utilizadas linguagens como o HTML (*Hypertext Markup Language*) que geram a estrutura do conteúdo da página e CSS (*Cascading Style Sheet*) que fará a estilização e formatação dos conteúdos visuais, como cores, fontes, tamanhos e layouts. Linguagens de programação como JavaScript podem ser utilizadas para adicionar interatividade e funcionalidade à interface, o que permite a validação de formulários, animações, manipulação de eventos e comunicação com o *back-end*.

O banco de dados é um sistema que permite o armazenamento, gerenciamento e recuperação dos dados de forma estruturada, atuando como um repositório centralizado e oferecendo recursos para manipulação, consulta e organização dos dados. É essencial para aplicações que necessitam de armazenamento e acesso eficiente para grande volume de dados. O sistema gerenciador de banco de dados utilizado foi o PostgreSQL, por ser relacional e de código aberto, oferecendo uma ampla gama de recursos avançados e sendo reconhecido por sua confiabilidade, escalabilidade e conformidade com os padrões SQL. Tem suporte para uma variedade de tipos dados, incluindo números, textos, data/hora e fornece suporte para recursos avançados como *ACID*, *triggers* e *stored procedures*.

3.1 ESTADO DA ARTE

No estado da arte, será apresentada uma solução já disponível no mercado que pode ser utilizada por equipes casuais para facilitar o agendamento de partidas com outras equipes.

3.1.1 WHATSAPP

De acordo com o WhatsApp (2023), mais de dois bilhões de pessoas em mais de 180 países utilizam o WhatsApp. O aplicativo começou como uma alternativa ao SMS e já oferece suporte ao envio e recebimento de diversos tipos de mídia. A ferramenta é gratuita e está disponível em qualquer lugar do mundo. Na Figura 4, é apresentada a interface de conversa em grupo.

Figura 4 – WhatsApp



Fonte: WhatsApp (2023)

O WhatsApp é uma das ferramentas disponíveis no mercado que oferece bate-papo entre usuários em tempo real, podendo auxiliar equipes a marcarem partidas casuais. Sua principal vantagem é ser amplamente conhecido e utilizado por diversas pessoas. No entanto, sua desvantagem é que só é possível iniciar uma conversa com quem você já tem o contato, tornando impossível procurar por outras equipes.

3.1.2 BUSCAQUADRAS

De acordo com o BuscaQuadras (2023), “O BuscaQuadras é um Portal onde o jogador/esportista pode realizar o agendamento online de horários nos Centros Esportivos de todo o Brasil, que utilizam o sistema SAQES para gestão do seu negócio”.

Continuando com BuscaQuadras (2023), “O jogador mensalista também pode acompanhar e pagar suas mensalidades, criar sua rede de amigos e ficar informado sobre todas as novidades do Centro Esportivo em que pratica seu esporte.”

A Figura 5 representa a página inicial do site oficial.

Figura 5 – BuscaQuadras

Encontre uma quadra perto de você!

Qual esporte? Onde? [Encontrar](#)

[Todas](#) AC AL AP AM BA CE DF ES GO MA MT MS MG PA PB PR PE PI RJ RN RS RO RR SC SP SE TO



Arena A10 Society
Rua Sargento Ananias, 571
Copa Cabana 3, Nova Viçosa - BA



Pequifutevolei
Goiania - GO



Chute Certo Rio Sport
Av Ayrton Sena, 2451
Tijuca, Rio De Janeiro - RJ



Centro Esportivo Pezzi
Porto Alegre - RS



Mcm Esportes
Av. Frederico Augusto Ritter, 1800
Dist. Industrial, Cachoeirinha - RS



Mcm Esportes Big
Rua Sergio Jungblut Dieterich, 1011
São João, Porto Alegre - RS



Stadium Futebol Society
Rua Bassano, 585
Sao Roque, Farroupilha - RS



Sarandi Esportes
Av Francisco Silveira Bitencourt, 1035
Sarandi, Porto Alegre - RS

1 2 3 4 5 6 7 8 9 10

Fonte: BuscaQuadras (2023)

4. MATCHSTART: PROTÓTIPO DE SOFTWARE PARA AGENDAMENTO DE PARTIDA PARA EQUIPES CASUAIS

Neste capítulo, serão abordados todos os processos executados referente ao desenvolvimento do protótipo deste trabalho.

4.1 ANÁLISE

Na revisão da literatura, foram apresentados temas relacionados à engenharia de software, essenciais para compreender as necessidades do projeto e atingir o objetivo final.

Com a aplicação de conceitos da engenharia de software foi possível analisar e identificar os processos necessários para o desenvolvimento do protótipo, principalmente com a análise de requisitos que ajudou a transformar as necessidades em requisitos de software claros e compreensíveis com aquilo que o trabalho propõe.

4.1.1 VISÃO GERAL

O MatchStart tem como objetivo facilitar a organização de partidas para equipes casuais, proporcionando recursos que simplifiquem a busca por outras equipes interessadas em disputar uma partida.

Ao acessar a aplicação o usuário deverá criar uma conta pessoal, onde ficarão salvos seus dados pessoais, como e-mail, senha e contato. Seguindo o processo de cadastro, o usuário terá que cadastrar sua primeira equipe, assim sendo possível acessar a parte principal do projeto. Cada usuário poderá ter até 3 equipes cadastradas.

Dentro da aplicação, o usuário poderá acessar o menu principal, onde ele poderá fazer a busca por outras equipes que desejem disputar uma partida e enviar convites para estas, as equipes do usuário também poderão receber convites de outras equipes, com os convites recebidos, as equipes deverão aceitar ou recusar, quando o convite é aceito o usuário receberá uma notificação com informações sobre o contato da outra equipe para facilitar a conversa entre ambos.

4.1.2 COMPARAÇÃO DO PROTÓTIPO COM O ESTADO DA ARTE

No quadro 2 é possível identificar as diferenças entre o protótipo do trabalho atual com o software descrito no estado da arte.

Quadro 2 – Comparação entre protótipo x estado da arte

Funcionalidade	WhatsApp	BuscaQuadras	MatchStart
Chats por mensagem, voz ou áudio	X		
Sistema filtra pelo local de atuação		X	X
Possibilita a busca por equipes praticantes do mesmo esporte			X
Aplicativo específico para esportes		X	X
Já possui reconhecimento	X	X	
Organização de partidas			X
Visualizar as partidas agendadas			X

Fonte: Acervo do Autor (2023).

4.2 REGRAS DE NEGÓCIO

As regras de negócio são essenciais para assegurar que o software atenda aos requisitos e necessidades, garantindo que o protótipo alcance seu propósito. No Quadro 3, serão listadas as principais regras de negócio relacionadas ao protótipo.

Quadro 3 – Regras de negócio

Número	Descrição
RN01	O usuário deve possuir o cadastro de pelo menos uma equipe para poder encontrar uma partida.
RN02	O usuário pode ter até três equipes cadastradas.
RN03	O sistema não deve permitir uma equipe possuir mais de uma partida no mesmo horário.
RN04	Quando o usuário fizer o login, o sistema deve logar automaticamente com uma equipe, a principal.
RN05	O sistema deve permitir ao usuário selecionar quantas cidade ele quiser para que seus times realizem partidas.
RN06	O usuário deve somente escolher um esporte para cada equipe cadastrada.
RN07	O usuário pode editar ou excluir os seus dados e dados da equipe sempre que quiser, mas não deverá ter partidas em andamento.
RN08	O sistema deve disponibilizar as equipes os seguintes esportes: Futsal, futebol de campo, futebol suíço e vôlei.
RN09	Por conta da LGPD (Lei Geral de Proteção de Dados), o usuário deverá aceitar os termos de uso para poder ter seus dados de telefone compartilhados com outras equipes.
RN10	O sistema deve permitir ao usuário criar e receber convites de partida.
RN11	O usuário deve receber uma notificação quando o time adversário aceitar um convite de partida ou recusar o mesmo.

Fonte: Acervo do Autor (2023).

4.3 REQUISITOS

Os requisitos funcionais são essenciais para descrever as funções, capacidades e comportamentos que o software deve possuir para atender o propósito do protótipo. No quadro 4, são descritos os principais requisitos funcionais referentes ao protótipo.

Quadro 4 – Requisitos funcionais

Número	Nome	Descrição	RN
RF01	Cadastro de Usuário	O sistema deve possuir uma rotina para criar o cadastro do usuário, tendo os campos de e-mail, senha e nome do usuário. Adicionalmente, por conta de regras de privacidade de dados em especial a LGPD, o usuário para poder ter seu número de telefone divulgado com outras equipes, deverá marcar uma check-box de aceite dos termos de uso.	RN01, RN09
RF02	Intervalo entre partidas	O sistema deve validar e proibir a uma equipe possuir um jogo no mesmo horário, ele não poderá aceitar nem criar convites de partidas em que o horário seja o mesmo de alguma já em andamento.	RN03
RF03	Editar Cadastro de Usuário	O sistema deve possuir uma rotina que permita ao usuário alterar os dados referente as suas equipes, como nome e esporte da equipe e senha e telefone do usuário	RN07
RF04	Excluir Usuário	O sistema deve possibilitar o usuário excluir sua conta.	RN07
RF05	Cadastro de Equipes	O sistema deve possuir uma rotina para que o usuário possa criar, alterar e remover as equipes, tendo campos como nome da equipe e esporte, com limite de 3 equipes por usuário.	RN01, RN02, RN06, RN08
RF06	Login de Equipe	O sistema deve possuir uma rotina para que o usuário faça a autenticação com sua equipe cadastrada.	RN04
RF07	Gerenciar Partidas	O sistema deve possibilitar ao usuário a encontrar outras equipes para realizar uma partida, enviar convites de partida para outras equipes, consultar os convites de partida recebidos, poder aceitar ou recusar os convites e visualizar as partidas em andamento.	RN10
RF08	Área de Notificações	O sistema deve notificar o usuário por meio de um alerta, quando um convite de uma de suas equipes for aceito ou recusado.	RN11
RF09	Locais de Atuação	O sistema deve possuir uma rotina para que o usuário adicione e remova as cidades na qual seus times disputarão.	RN05

Fonte: Acervo do Autor (2023).

Os requisitos não funcionais descrevem as qualidades e características que um software deve possuir, mas que não estão relacionados às funcionalidades específicas que o sistema deve ter, abordando aspectos como desempenho,

confiabilidade e segurança. No quadro 5, são apresentados os principais requisitos não funcionais do protótipo.

Quadro 5 – Requisitos não funcionais

Número	Nome	Descrição
RNF01	Banco de dados	O banco de dados deve usar como SGDB o PostgreSQL
RNF02	Linguagem <i>Front-end</i>	Deve ser utilizado o ReactJs, biblioteca <i>front-end</i> baseado em Javascript.
RNF03	Linguagem <i>Back-end</i>	Deve ser utilizado o NodeJs, biblioteca <i>back-end</i> baseado em Javascript.
RNF04	Senhas de Usuário	Qualquer senha deve ser criptografada antes de serem inseridas ao banco de dados e devem possuir pelo menos 8 caracteres.
RNF05	Responsividade	As telas do protótipo devem ser responsivas, adaptando-se aos dispositivos com resolução mínima de 390 x 960.

Fonte: Acervo do Autor (2023).

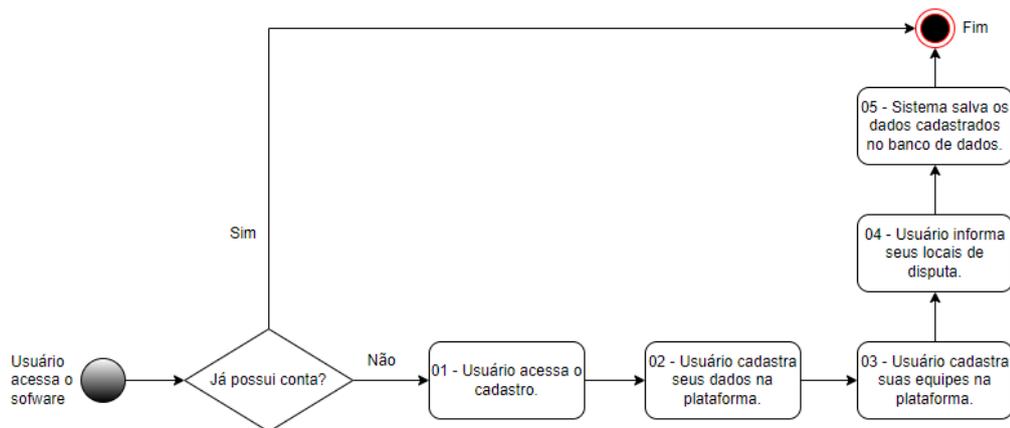
4.4 DIAGRAMAS

Nesta seção serão listados os diagramas referentes aos processos de utilização do sistema e as ações dos usuários logados. O objetivo é melhorar o entendimento dos fluxos e funcionamento do protótipo.

O usuário que desejar utilizar dos processos do software deverá iniciar cadastrando seus dados na plataforma, após isto será necessário fazer o cadastro de pelo menos uma de suas equipes e informar seus locais de atuação. Com isto o usuário terá acesso à tela inicial, onde ele poderá gerenciar suas equipes e fazer a busca por outras equipes que desejarem realizar uma partida casual.

Observando a Figura 6, é possível verificar o fluxograma do processo de cadastro de usuário.

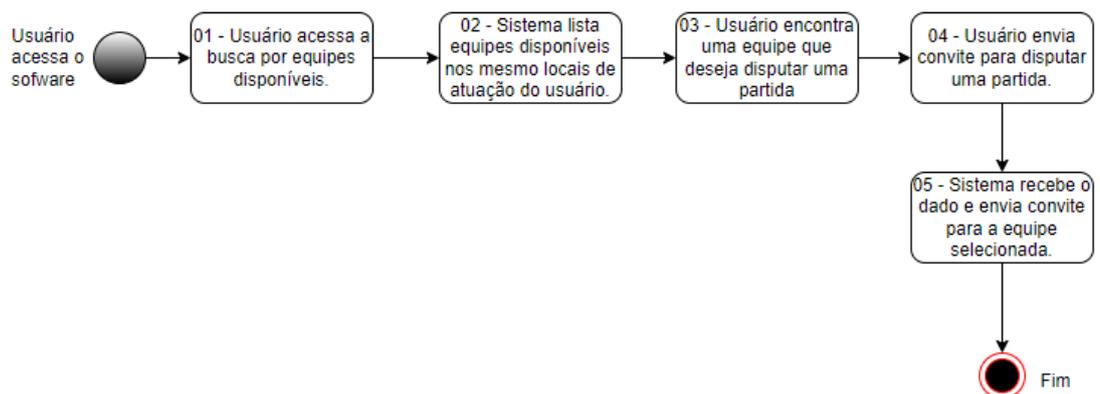
Figura 6 – Fluxograma do processo de cadastro de usuário



Fonte: Acervo do Autor (2023).

Para realizar a busca por alguma equipe para realizar uma partida casual, o usuário deverá acessar a busca por equipes disponíveis, onde o sistema irá realizar uma busca (atividade 01) no banco de dados e fará uma lista com as equipes compatíveis com os locais de atuação do usuário (atividade 02). O usuário deverá enviar um convite para as equipes que ele quiser realizar uma partida (atividade 04), o sistema receberá os convites e fará o envio dos dados para as equipes selecionadas (atividade 05). É possível visualizar este processo no fluxograma apresentado na Figura 7.

Figura 7 – Fluxograma para buscar por equipes disponíveis

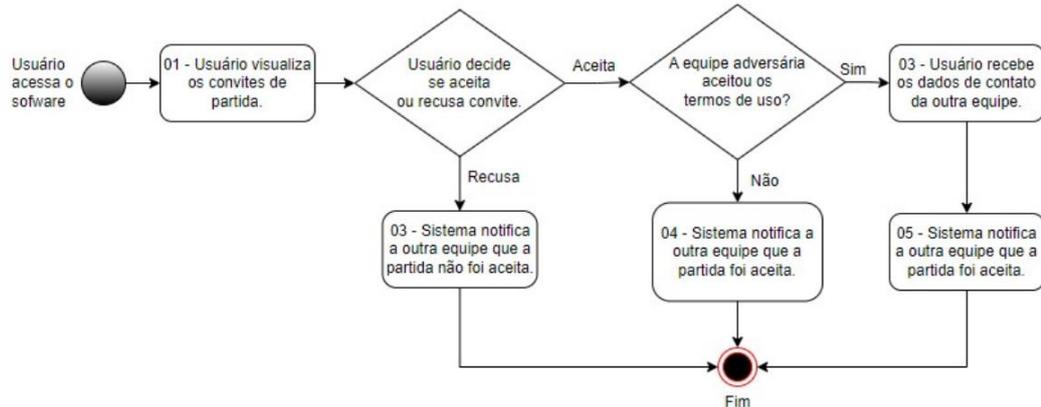


Fonte: Acervo do Autor (2023).

Quando receber um convite para realizar uma partida, o usuário deverá aceitar ou recusar. Caso o usuário recuse o convite, o sistema somente irá notificar a outra equipe que o convite foi recusado (atividade 03), mas caso o usuário aceite o convite,

o sistema verificará se o adversário aceitou os termos de uso do software. Caso sim, o usuário irá receber os dados de contato da outra equipe (atividade 02) e o sistema notificará a outra equipe que a partida foi aceita (atividade 05), caso não, somente irá notificar a outra equipe sobre a confirmação (atividade 04). Todas as atividades podem ser visualizadas no fluxograma da Figura 8.

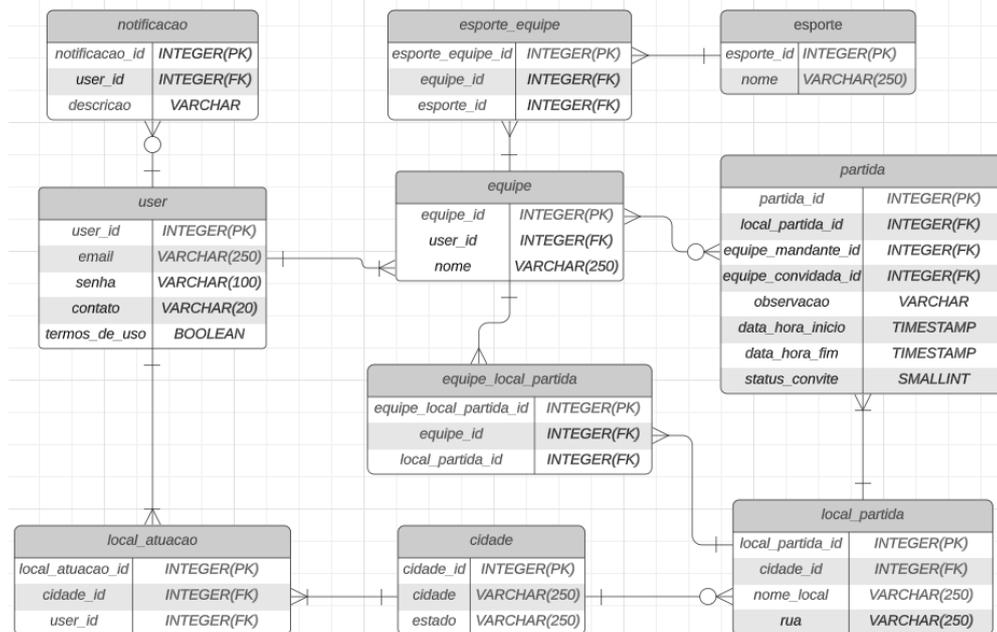
Figura 8 – Fluxograma para visualizar convites de partidas



Fonte: Acervo do Autor (2023).

A Figura 9 representa o Diagrama de Entidade e Relacionamento (DER) do protótipo, cujo objetivo principal é demonstrar as tabelas e os relacionamentos presentes no banco de dados do projeto.

Figura 9 – Diagrama de entidade e relacionamento



Fonte: Acervo do Autor (2023).

4.5 IMPLEMENTAÇÃO

O protótipo teve como base para o desenvolvimento a utilização do JavaScript, para o *front-end* foi utilizado da biblioteca ReactJs, que tem como objetivo facilitar a criação de interfaces web. Na Figura 10 está representado a criação de funções utilizando o ReactJs.

Figura 10 – Criação de funções com ReactJs

```
import React, { useState, useEffect } from 'react'; ...

const Convites = () => {

  const [convites, setConvites] = useState([]);
  const [atualizaConvites, setAtualizaConvites] = useState();
  const [celulaExpandida, setCelulaExpandida] = useState(null);
  const [openRecusado, setOpenRecusado] = useState(false);
  const [openAceito, setOpenAceito] = useState(false);
  const [open2, setOpen2] = useState(false);

  const handleExpansaoCelula = (index) => {
    if (celulaExpandida === index) {
      setCelulaExpandida(null);
    } else {
      setCelulaExpandida(index);
    }
  };

  const handleClose = (reason) => {
    if (reason === 'clickaway') {
      return;
    }
    setOpenRecusado(false);
    setOpenAceito(false);
    setOpen2(false);
  };

  // Busca os Convites de Partidas
  useEffect(() => {
    const data = {
      equipe_id: localStorage.getItem("equipe_id")
    }
  });
};
```

Fonte: Acervo do Autor (2023).

Para a renderização dos componentes na interface foram definidas rotas, a organização das rotas está representada na Figura 11.

Figura 11 – Organização de Rotas da Interface

```
function AppRoutes() {
  const location = useLocation();

  const isCadastroRoute = location.pathname === '/cadastro';
  const isLoginRoute = location.pathname === '/login';

  return (
    <div className="App">
      {(!isCadastroRoute && !isLoginRoute) && <Sidebar />}
      <Routes>
        <Route path="/" element={<Inicio />} />
        <Route path="/notificacao" element={<Notificacao />} />
        <Route path="/convites" element={<Convites />} />
        <Route path="/atuacao" element={<Settings />} />
        <Route path="/equipes" element={<Equipes />} />
        <Route path="/partidas" element={<Partidas />} />
        <Route path="/configuracoes" element={<Usuario />} />
        <Route path="/cadastro" element={<Cadastro />} />
        <Route path="/login" element={<Login />} />
      </Routes>
    </div>
  );
}
```

Fonte: Acervo do Autor (2023).

Para auxiliar na criação das interfaces, foi utilizado do Material-UI, que é uma biblioteca de componentes de interface de usuário para o React. Possui componentes fáceis de personalizar que auxiliam nas necessidades específicas do design de um aplicativo. A Figura 12 representa o uso e importação do Material-UI.

Figura 12 - Utilização do Material-UI

```
import { Grid, Button, Card, CardContent, Typography, TextField, MenuItem } from '@material-ui/core';

return (
  <Card>
    <CardContent>
      <Typography variant="h5" component="p" align="center">
        Editar Equipe
      </Typography>
      <TextField>
        Label="Nome da Equipe"
        fullWidth
        margin="normal"
        value={nome}
        onChange={(e) => setNome(e.target.value)}
      </TextField>
      <TextField>
        Label="Esporte"
        fullWidth
        margin="normal"
        select
        value={esporte}
        onChange={(e) => setEsporte(e.target.value)}
      >
        {esportes.map((option) => (
          <MenuItem key={option.value} value={option.value}>
            {option.label}
          </MenuItem>
        ))}
      </TextField>
      <Grid container justify="center" style={{ marginTop: '20px' }}>
        <Button variant="contained" color="primary" style={{ marginRight: '10px' }} onClick={handleSalvar}>
          Salvar
        </Button>
        <Button variant="contained" onClick={onCancel}>
          Cancelar
        </Button>
      </Grid>
    </CardContent>
  </Card>
);
```

Fonte: Acervo do Autor (2023).

A conexão do *front-end* com o *back-end* ocorre por meio da utilização do Axios, que é uma biblioteca JavaScript utilizada para fazer requisições HTTP a servidores. A Figura 13 representa a utilização do Axios para gerar o arquivo “api.js” que faz a conexão com o servidor.

Figura 13 - Utilização do Axios

```
JS api.js ×
src > JS api.js > ...
1   import axios from "axios";
2
3   const api = axios.create({
4     |   baseURL: 'http://localhost:3000/'
5   });
6
7
8   export default api;
```

Fonte: Acervo do Autor (2023).

Para o back-end, foi utilizado o ambiente de execução JavaScript denominado 'Node.js', que permite a utilização do JavaScript no lado do servidor e a criação da API REST. Para facilitar o desenvolvimento, foi utilizado o framework Express, que auxilia na definição de rotas para a manipulação de requisições HTTP. A Figura 14 representa a criação das rotas principais utilizando o Express.

Figura 14 - Rotas com Express

```
const express = require('express');
const app = express();
const port = 3000;
const cors = require('cors');

// Rotas
const index = require('./routes/index');
const cadastroRoute = require('./routes/cadastro.routes');
const esporteRoute = require('./routes/esportes.routes');
const cidadeRoute = require('./routes/cidades.routes');
const loginRoute = require('./routes/login.routes');
const localAtuacaoRoute = require('./routes/localAtuacao.routes');
const equipesRoute = require('./routes/equipes.routes');
const partidaRoute = require('./routes/partida.routes');
const notificacoesRoute = require('./routes/notificacoes.routes');

app.use(express.json({ limit: '50mb' }));
app.use(express.urlencoded({ limit: '50mb', extended: true }));
app.use(express.json());
app.use(express.json({ type: 'application/json' }));

// Importações
app.use(cors());
app.use(index);
app.use('/cadastro', cadastroRoute);
app.use('/esportes', esporteRoute);
app.use('/cidades', cidadeRoute);
app.use('/login', loginRoute);
app.use('/atuacao', localAtuacaoRoute);
app.use('/equipes', equipesRoute);
app.use('/partida', partidaRoute);
app.use('/notificacao', notificacoesRoute);
```

Fonte: Acervo do Autor (2023).

Com a utilização das ferramentas apresentadas foi possível gerar um protótipo responsivo, com interfaces atrativas e padronizadas.

4.6 UTILIZAÇÃO E FUNCIONAMENTO

O desenvolvimento das interfaces do protótipo teve como base os requisitos funcionais, utilizando o framework JavaScript denominado React, que facilita a construção de interfaces de usuário interativas e dinâmicas.

O protótipo foi desenvolvido de forma responsiva, possibilitando sua visualização em dispositivos web e móveis. Um exemplo disso pode ser observado na Figura 15, onde a tela de cadastro da pessoa é exibida em formato responsivo para telas com resoluções de 390 x 960 até 1920 x 1080.

Figura 15 – Tela de login Mobile

A imagem mostra a tela de login mobile do sistema MatchStart. No topo, o título "MatchStart" é exibido em uma fonte grande e preta. Abaixo dele, o subtítulo "PROTÓTIPO DE SOFTWARE PARA AGENDAMENTO DE PARTIDA PARA EQUIPES CASUAIS" aparece em uma fonte menor e em letras maiúsculas. O formulário de login contém dois campos de entrada: "Email *" e "Senha *", ambos com bordas arredondadas e um ícone de olho para alternar a visibilidade da senha. Abaixo dos campos, há um botão verde com o texto "ENTRAR" em branco. Na base da tela, o texto "Não Está Cadastrado? [Criar Conta](#)" é exibido, com "Criar Conta" sendo um link azul sublinhado.

Fonte: Acervo do Autor (2023).

4.6.1 Tela de Login do Usuário

Ao acessar o sistema, o usuário se encontrará com a tela de login, onde será realizada a autenticação utilizando do e-mail e senha, o sistema fará automaticamente o login com alguma das equipes do usuário (RN04). A Figura 16 demonstra a tela de login, tanto web quanto em mobile.

Figura 16 – Tela de loginA imagem mostra a tela de login do sistema MatchStart. No topo, o título "MatchStart" é exibido em uma fonte grande e preta. Abaixo dele, o subtítulo "PROTÓTIPO DE SOFTWARE PARA AGENDAMENTO DE PARTIDA PARA EQUIPES CASUAIS" aparece em uma fonte menor e em letras maiúsculas. O formulário de login contém dois campos de entrada: "Email *" e "Senha *", ambos com bordas arredondadas e um ícone de olho para alternar a visibilidade da senha. Abaixo dos campos, há um botão verde com o texto "ENTRAR" em branco. Na base do formulário, o texto "Não Está Cadastrado? [Criar Conta](#)" é exibido, com "Criar Conta" sendo um link azul sublinhado.

Fonte: Acervo do Autor (2023).

Caso o usuário ainda não possua suas credenciais de login, ele deverá acessar a opção “Criar conta”, para acessar a interface de cadastro.

4.6.2 - Tela de Cadastro de Usuário

A Figura 17 demonstra a tela de cadastro (RF01), tanto web quanto em mobile. Nesta interface o usuário deverá criar sua conta que ele utilizará para realizar a autenticação no sistema.

O usuário também deverá selecionar o seu primeiro local de atuação que são dois inputs, o primeiro de “Estado” que ao ser selecionado informa todos os estados brasileiros, ao selecionar o estado desejado é enviado uma requisição ao *back-end* que realiza uma busca no banco de dados por todas as cidades e faz a liberação do input “Cidade” que apresentará as cidades referentes ao estado selecionado anteriormente, e por fim criar a sua primeira equipe, desta forma ele já fará o acesso ao sistema com os requisitos necessários para utilização do sistema (RN01).

Caso o usuário rejeite os termos de uso, o seu número de contato não será compartilhado entre as outras equipes (RN09)

Figura 17 – Tela de cadastro

MATCHSTART
Cadastro De Usuário

Email *

Senha *

Confirmar Senha *

Telefone *

Local De Atuação:
Após a finalização do cadastro você poderá alterar e inserir mais locais de atuação.

Estado * Cidade *

Primeira Equipe:

Nome Da Primeira Equipe *

Esporte *

Termo de Uso: Ao marcar este checkbox o usuário aceita que seu número seja compartilhado com outras equipes para facilitar a comunicação.

CADASTRAR

Já Tem Uma Conta? [Login](#)

Fonte: Acervo do Autor (2023).

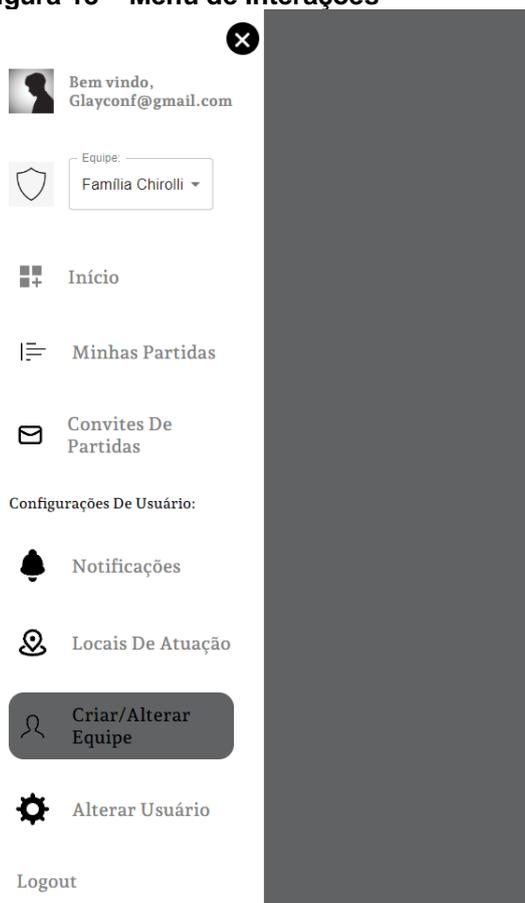
Com o cadastro criado, o usuário será redirecionado para a interface principal do sistema.

4.6.3 Menu de Interações

A Figura 18 demonstra o menu de interações do protótipo, onde o usuário poderá acessar as rotinas do protótipo e realizar a autenticação com suas equipes.

Neste menu o usuário poderá acessar as rotinas para a equipe autenticada no input “Equipe” e as configurações do seu usuário.

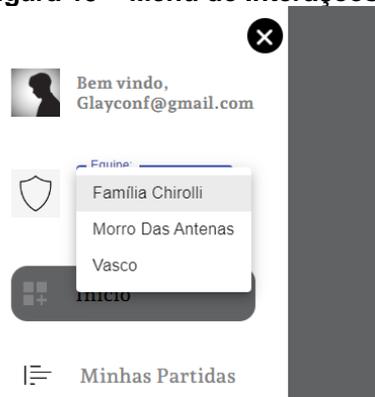
Figura 18 – Menu de Interações



Fonte: Acervo do Autor (2023).

Destacado na Figura 19, o menu de interações possibilitará a autenticação entre as equipes (RF06). Ao selecionar o Input “Equipe”, o usuário visualizará suas equipes criadas. As fotos de usuário e time são padrões para todos.

Figura 19 – Menu de Interações Input de Equipe



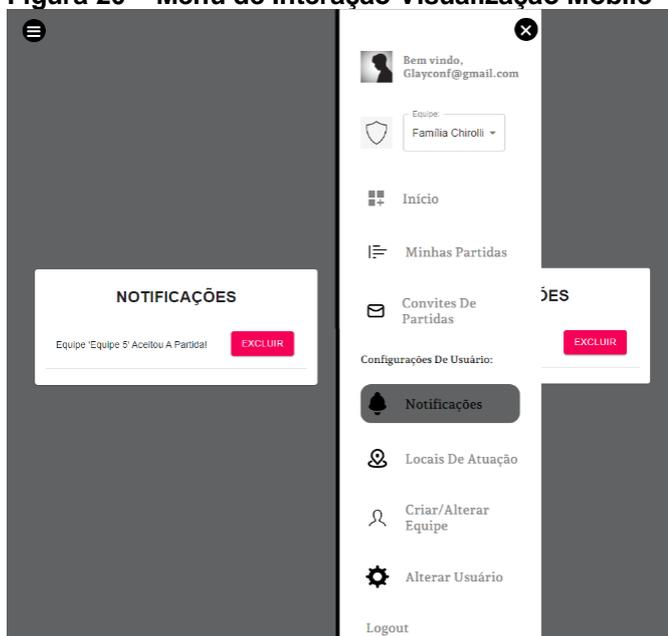
Fonte: Acervo do Autor (2023).

A autenticação entre equipes tem como objetivo organizar partidas nas rotinas específicas relacionadas às equipes. Por exemplo, ao acessar a interface 'Convites

de Partidas', o usuário visualizará apenas os convites de partida referentes à equipe autenticada. Ao realizar a autenticação do usuário, o sistema fará automaticamente a autenticação com uma das equipes.

Na Figura 19, é possível observar um botão em forma de 'X', que permite recolher o menu de interações. Essa função visa facilitar a visualização das interfaces para usuários em dispositivos móveis. A Figura 20 compara a visualização da interface 'Notificações' em um dispositivo móvel com o menu de interações recolhido e estendido.

Figura 20 – Menu de Interação Visualização Mobile



Fonte: Acervo do Autor (2023).

4.6.4 Interface de Início

A interface “Início” será a primeira tela que o usuário visualizará ao realizar a autenticação, ela possibilita o usuário a visualizar as partidas em andamento referente a equipe autenticada. Representa uma parte do RF07.

O usuário tem duas opções de visualização. A primeira são as partidas 'em casa', que são aquelas criadas pelo usuário e já aceitas pelo usuário convidado. A segunda opção são as partidas 'fora de casa', que consistem nos convites de partida recebidos de outros usuários e já aceitos.

Nas duas opções de visualização é possível o usuário verificar as partidas que estão para ocorrer, somente serão listadas aquelas partidas que estão ocorrendo ou

irão ocorrer, por exemplo, se a partida é no dia 25/07/2023 das 14h até as 15h, quando chegar no dia 25/07/2023 as 15h e 01 minutos o sistema não listará mais a partida.

Na Figura 21, é possível visualizar a interface demonstrando as partidas 'em casa' para a equipe 'Família Chirolli'.

Figura 21 – Interface de Início Partidas Em Casa

The screenshot shows a web interface titled 'PARTIDAS EM ANDAMENTO'. On the left, there is a blue button labeled 'PARTIDAS FORA'. The main content area is titled 'Partidas Em Casa' and contains a table with the following data:

Adversário	Cidade	Rua	Local	Hora Início	Hora Fim
> Equipe 5	Laurentino	Rua Dos Vereadores	Campo Do Local	2023-11-11 14:23:00	2023-11-11 15:23:00

Fonte: Acervo do Autor (2023).

Na Figura 22, é possível visualizar a interface demonstrando as partidas fora de casa para a equipe "Família Chirolli".

Figura 22 – Interface de Início Partidas Fora de Casa

The screenshot shows a web interface titled 'PARTIDAS EM ANDAMENTO'. On the left, there is a blue button labeled 'PARTIDAS EM CASA'. The main content area is titled 'Partidas Fora De Casa' and contains a table with the following data:

Adversário	Cidade	Rua	Local	Hora Início	Hora Fim
▼ Equipe 5	Rio Do Sul	Rua Quatro	Bar Do Zé	2023-11-10 13:31:00	2023-11-10 14:31:00

Below the table, there are two sections:

- Contato:** Contato Não Divulgado
- Observação:** Chegar Cedo

Fonte: Acervo do Autor (2023).

Nas tabelas apresentadas na interface 'Início', ao clicar em seus registros, é possível visualizar o 'Contato' do adversário e as 'Observações' para a partida.

O contato do usuário da equipe adversária só será informado se este tiver aceitado os termos de uso do protótipo; caso contrário, será exibida a mensagem

'Contato não divulgado'. Essa abordagem respeita a Lei Geral de Proteção de Dados e representa uma parte do RF01. Isso é demonstrado na Figura 23.

Figura 23 – Interface de Início Contato e Observação

	Adversário	Cidade	Rua	Local	Hora Início	Hora Fim
▼	Equipe 5	Rio Do Sul	Rua Quatro	Bar Do Zé	2023-11-10 13:31:00	2023-11-10 14:31:00
Contato: Contato Não Divulgado			Observação: Chegar Cedo			

Fonte: Acervo do Autor (2023).

4.6.5 Interface de Minhas Partidas

A interface 'Minhas Partidas' têm como objetivo oferecer ao usuário a opção de criar convites de partidas (RN10) e visualizar os convites que ainda não foram aceitos ou recusados. As informações que aparecem nesta tela são referentes a equipe que o usuário está autenticado. Representa parte do RF07 e RF02.

Na Figura 24, é possível visualizar a criação de um convite de partida. Para criar um convite o usuário deverá inserir os dados referentes ao local onde ocorrerá a partida, observação, hora de início e fim da partida e equipe para qual o convite será enviado. Para escolher a equipe que será convidada a disputar uma partida o usuário deverá primeiro selecionar a cidade da partida, desta forma o sistema por meio da API fará uma busca no banco de dados e filtrará todas com as equipes que disputam o mesmo esporte que a equipe autenticada e atuam na cidade selecionada pelo usuário para criar o convite de partida. Caso o usuário insira uma data que o mesmo já possua uma partida em andamento ou com o convite pendente, o sistema irá informar um alerta pedindo para o usuário alterar o horário da partida e uma mensagem em vermelho informando que o usuário já possui uma partida neste horário (RN03).

Figura 24 – Interface de Minhas Partidas Criar convite de partida

Fonte: Acervo do Autor (2023).

Na Figura 25 é possível visualizar os convites pendentes, caso a equipe adversária não tenha aceitado ainda o convite de partida o usuário poderá cancelar o convite, desta forma ele disponibilizará o horário para gerar um novo convite.

Figura 25 – Interface de Minhas Partidas Convites Pendentes

Adversário	Cidade	Local	Rua	Observação	Início	Fim	Ações
Equipe Sete	Laurentino	Campo Do Santos	Rua Dos Vereadores	A Entrada Fica Após Uma Lagoa Grande Na Direita.	2023-11-02 18:43:00	2023-11-02 19:43:00	CANCELAR
Equipe 5	Laurentino	Campo Do Local	Rua Treze	Chegar Minutos Antes	2023-11-02 20:43:00	2023-11-02 21:43:00	CANCELAR

Fonte: Acervo do Autor (2023).

4.6.6 Interface dos Convites de Partidas

A interface 'Convites de Partidas' tem como objetivo oferecer ao usuário as opções de aceitar ou recusar partidas (RN10). Ao acessar a interface, serão apresentados ao usuário os convites de partidas recebidos para a equipe autenticada, proporcionando as opções de 'Aceitar' e 'Recusar'. Ao clicar em alguma célula da tabela, a 'Observação' referente à partida é apresentada. Nesta interface, está representada parte do RF07 e RF02. Na Figura 25, é possível visualizar a interface mencionada.

Figura 26 – Interface Convites de Partidas

CONVITES RECEBIDOS							
	Adversário	Cidade	Rua	Local	Hora Início	Hora Fim	
∨	Equipe 5	Laurentino	Rua Quatro	Campo Do Zê	2023-11-04 14:53:00	2023-11-04 15:53:00	ACEITAR RECUSAR
Observação: Trazer Água							
>	Equipe 10	Laurentino	Rua 4	Campo Centro	2023-11-04 14:55:00	2023-11-04 15:55:00	ACEITAR RECUSAR

Fonte: Acervo do Autor (2023).

Se o usuário tentar aceitar um convite em que já exista outra partida em andamento no mesmo horário, o sistema não permitirá a aceitação e emitirá um alerta informando que o convite de partida não pode ser aceito. Isso é demonstrado na Figura 27.

Figura 27 – Interface Convites de Partidas Alerta

CONVITES RECEBIDOS							
	Adversário	Cidade	Rua	Local	Hora Início	Hora Fim	
>	Equipe 10	Laurentino	Rua 4	Campo Centro	2023-11-04 14:55:00	2023-11-04 15:55:00	ACEITAR RECUSAR

Fonte: Acervo do Autor (2023).

4.6.7 Interface das Notificações

A interface 'Notificações' representa o RF08 e refere-se às informações do usuário, não das equipes. As notificações serão relacionadas a todos os convites aceitos ou recusados, independentemente da equipe autenticada. Caso não seja mais do interesse do usuário, ele poderá excluir a notificação (RN11). A Figura 28 apresenta a tela de notificações.

Figura 28 – Interface de Notificações



Fonte: Acervo do Autor (2023).

4.6.8 Interface dos Locais de Atuação

Representando o RF09, a interface 'Locais de Atuação' tem como objetivo oferecer ao usuário a opção de adicionar e remover cidades onde ele disputará partidas, isto afetará diretamente nos convites de partidas que o usuário irá receber e enviar (RN05).

O Input de cidade é bloqueado, quando o usuário selecionar o estado o sistema realizará uma busca no banco de dados por meio de uma requisição na api que retornará todas as cidades referentes aquele estado e fará a liberação do campo de cidade para o usuário. A Figura 29 representa a interface 'Locais de Atuação'.

Figura 29 – Interface de Locais de Atuação

Adicione e remova as cidades que suas equipes disputam partidas.

Estado Cidade

ADICIONAR CIDADE

Estado	Cidade	Ações
SC	Rio Do Sul	REMOVER
SC	Laurentino	REMOVER
SC	Rio Do Oeste	REMOVER

Fonte: Acervo do Autor (2023).

4.6.9 Interface para Criar/Alterar Equipes

Para utilizar o sistema o usuário deve conter pelo menos uma equipe (RN01), que é criada no cadastro do usuário, a interface 'Criar/Alterar Equipe' representando o RF05, tem como objetivo permitir a alteração e criação de novas equipes. Cada equipe poderá ter um esporte cadastrado (RN06), dentre as opções futsal, futebol de campo, futebol suíço e vôlei (RN08). A Figura 30 representa a interface.

Figura 30 – Interface Criar/Alterar Equipe

Crie, altere e remova as suas equipes.
Você pode ter até 3 equipes.

Nome Da Equipe Esporte

ADICIONAR EQUIPE

Nome Da Equipe	Esporte	Ações
Família Chirolli	Futebol De Campo	ALTERAR REMOVER
Vasco	Futebol De Campo	ALTERAR REMOVER

Fonte: Acervo do Autor (2023).

Cada usuário poderá ter até 3 equipes cadastradas no sistema (RN02), ao atingir o limite os campos 'Nome da Equipe', 'Esporte' e o botão 'Adicionar Equipe' não serão mais apresentados na interface. Isto está representado na Figura 31.

Figura 31 – Interface Criar/Alterar Equipe 2

Crie, altere e remova as suas equipes.
Você pode ter até 3 equipes.

Nome Da Equipe	Esporte	Ações
Família Chirolli	Futebol De Campo	ALTERAR REMOVER
Morro Das Antenas	Futsal	ALTERAR REMOVER
Vasco	Futebol De Campo	ALTERAR REMOVER

Fonte: Acervo do Autor (2023).

Não será permitido ao usuário remover uma equipe que esteja vinculada a uma partida em andamento ou a algum convite pendente de partida, caso o usuário tente

fazer a remoção o sistema retornará um alerta informando que não é possível (RN07). Esta ação é representada na Figura 32.

Figura 32 – Interface Criar/Alterar Equipe Alerta

Esta equipe está vinculada a uma partida ou convite pendente! Finalize a partida ou cancele os convites para remover esta equipe.

Crie, altere e remova as suas equipes.
Você pode ter até 3 equipes.

Nome Da Equipe	Esporte	Ações
Família Chirolli	Futebol De Campo	<button>ALTERAR</button> <button>REMOVER</button>
Morro Das Antenas	Futsal	<button>ALTERAR</button> <button>REMOVER</button>
Vasco	Futebol De Campo	<button>ALTERAR</button> <button>REMOVER</button>

Fonte: Acervo do Autor (2023).

Ao selecionar o botão 'Alterar', a interface irá abrir a opção 'Editar Equipe' e possibilitará ao usuário realizar a alteração do nome e esporte da equipe selecionada. A Figura 33 demonstra a interface de edição que representa parte do RF03.

Figura 33 – Interface Criar/Alterar Equipe Edição

Editar Equipe

Nome Da Equipe
Família Chirolli

Esporte
Futebol De Campo

SALVAR CANCELAR

Fonte: Acervo do Autor (2023).

4.6.10 Interface para Alterar Usuário

A alteração do usuário faz parte referência a parte do RF03, nesta tela o usuário poderá fazer a alteração de sua senha, telefone, temos de uso referente ao uso do seu contato e por fim, faz representa o RF04 possibilitando que o usuário delete sua conta.

Esta interface tem como objetivo garantir ao usuário, além de poder alterar sua senha e seu telefone, ter a chance de voltar atrás caso queira que seu contato seja divulgado com outras equipes para facilitar a comunicação entre ambas.

A Figura 34 representa a interface de alteração de usuário.

Figura 34 – Interface Alterar Usuário

A interface 'Alterar Usuário' apresenta o seguinte layout:

- Um formulário com o título 'Alterar Usuário' no topo.
- Três campos de entrada de texto para: 'Senha Atual *', 'Nova Senha *' e 'Confirmar Nova Senha *'.
- Um botão azul com o texto 'ALTERAR SENHA'.
- Um campo de entrada de texto para 'Telefone *' com o exemplo '(99) 99999-9999'.
- Um checkbox selecionado com o texto: 'Termo de Uso: Ao marcar este checkbox o usuário aceita que seu número seja compartilhado com outras equipes para facilitar a comunicação.'
- Um botão azul com o texto 'ALTERAR TELEFONE'.
- Um botão rosa com o texto 'DELETAR CONTA'.

Fonte: Acervo do Autor (2023).

Caso o usuário não queira mais utilizar do sistema e queira que seus dados sejam excluídos, ele poderá deletar sua conta selecionando o botão 'Deletar Conta' e realizando a confirmação final.

5. CONCLUSÃO

O presente trabalho teve como objetivo demonstrar um protótipo de software voltado a equipes casuais que desejam realizar uma partida de algum esporte coletivo, facilitando a procura por outras equipes na região utilizando de seus dispositivos pessoais conectados à rede.

Com a definição das regras de negócio foi possível identificar as necessidades do projeto para que fosse possível alcançar o intuito do protótipo. Desta forma foi possível identificar as necessidades reais que uma equipe precisa para realizar o agendamento de uma partida por meio de um software.

O levantamento de requisitos proporcionou a possibilidade de escrever as principais funções, comportamentos, entender e definir as necessidades e capacidades do protótipo. Tendo os requisitos descritos, o desenvolvimento do protótipo se tornou muito mais prático e ágil, pois deixou os objetivos e metas claros, proporcionando uma direção para o projeto.

Para o desenvolvimento do protótipo foram necessárias duas aplicações, uma para *front-end* e outra para *back-end*, sendo ambas desenvolvidas utilizando-se de ferramentas que possuem como base a linguagem de programação Javascript.

O *front-end* da aplicação foi desenvolvido utilizando da ferramenta ReactJS, esta que junto de suas bibliotecas permitiu uma fácil padronização das interfaces do protótipo e permitiu que o mesmo fosse desenvolvido de forma responsiva seguindo os requisitos e regras de negócio descritos.

O *back-end* do protótipo (servidor) foi desenvolvido utilizando da ferramenta NodeJS, com auxílio do *framework* Express, que permitiu o desenvolvimento da API REST sem muita complexidade, possibilitando gerenciar as requisições HTTP de forma simples e prática.

Ao realizar a pesquisa referente a ferramentas com objetivos semelhantes foram encontradas somente ferramentas com o intuito de realizar o aluguel de quadras e espaços para equipes, diferente do objetivo do presente trabalho que é facilitar a busca por outras equipes casuais que desejam disputar uma partida amadora, em diferentes locais.

O desenvolvimento e planejamento do protótipo se tornou muito mais fácil e ágil utilizando das ferramentas da engenharia de software, a aplicação cumpriu seus objetivos possibilitando que equipes que se cadastrem na ferramenta possam

encontrar e enviar convites para outras equipes que desejem realizar uma partida casual.

Os objetivos do trabalho foram alcançados e concluídos com sucesso. Considerando as limitações para a criação deste trabalho, a seguir serão listadas algumas novas funcionalidades e melhorias para trabalhos futuros.

5.1 RECOMENDAÇÕES DE TRABALHOS FUTUROS

Para trabalhos futuros seria recomendado gerar melhorias no software, para Android transformaria em um PWA (aplicativos PWA são acessados através de navegadores web, mas podem ser instalados na tela inicial do dispositivo, proporcionando uma experiência mais próxima à de aplicativos tradicionais.), e para iOS, seria adicionado um aplicativo mobile.

Para melhorar a comunicação, adicionar um chat entre as equipes, desta forma os usuários poderiam entrar em contato diretamente pelo software, sem a necessidade de utilizar de terceiros.

Com a afinidade de tornar a experiência do usuário melhor em relação as suas equipes, adicionar uma funcionalidade onde as equipes possam ter no seu cadastro o nome dos jogadores, além disso, adicionar a posição e estatísticas de cada usuário referente ao esporte da equipe.

Também é possível adicionar uma funcionalidade que permita as equipes ao finalizar uma partida possam adicionar o placar final destas partidas, indicando os goleadores, jogadores que deram assistências e avaliar a equipe adversária. Com a adição do placar, é possível adicionar um histórico de partidas da equipe e gerar estatísticas referente a equipe, assim o usuário terá a possibilidade de avaliar o desempenho de suas equipes.

Por fim, para engajar mais o uso do aplicativo, seria possível criar um site que divulgue a aplicação e eventos, também adicionar um ranking de equipes por esporte, possibilitando filtrar por cidade, região, estado e até mesmo nacional. Desta forma trazendo um pouco de competitividade as equipes que utilizam do sistema.

REFERÊNCIAS

ALVES, William Pereira. **Banco de dados: teoria e desenvolvimento**. 2. São Paulo Erica 2020.

ALVES, William Pereira. **HTML & CSS: aprenda como construir páginas web**. São Paulo Expressa 2021.

ALVES, William Pereira. **Linguagem e lógica de programação**. São Paulo Erica 2014.

BATISTA, Rafael. **Esportes**. 2023. Disponível em: <<https://brasilecola.uol.com.br/educacao-fisica/esportes.htm>>. Acesso em: 17 jun. 2023.

BUSCAQUADRAS. **BuscaQuadras: Encontre, Reserve e JOGUE!**. 2023. Disponível em: <<https://buscaquadras.com.br/>>. Acesso em: 04 dez. 2023.

CABLE NEWS NETWORK BRASIL. **Brasil tem mais smartphones que habitantes, aponta FGV**. 2022. Disponível em: <<https://www.cnnbrasil.com.br/economia/brasil-tem-mais-smartphones-que-habitantes-aponta-fgv/>>. Acesso em: 17 jun. 2023.

CARDOSO, Leandro da C. **Frameworks back end**. São Paulo Platos Soluções Educacionais 2021.

CARDOSO, Virginia M. **Linguagem sql: fundamentos e práticas**. São Paulo Saraiva 2009.

DIAS, Ariel da Silva. **Administração de bancos de dados**. São Paulo Platos Soluções Educacionais 2021.

FARIA, Marcel R. **DevOps para Infraestrutura de Redes**. 2017.

FITZGERALD, John; ODA, Tomohiro; MACEDO, Hugo Daniel. **Proceedings of the 18th International Overture Workshop**. ArXiv preprint arXiv:2101.07261, 2021.

FLANAGAN, David. **JavaScript: o guia definitivo**. 6. Porto Alegre Bookman 2014.

GACKENHEIMER, Cory. **Introduction to React**. Apress, 2015.

INSTITUTO VISÃO. **Diagnóstico Nacional do Esporte**: Diesporte. 2015.

MACHADO, Felipe Nery Rodrigues. **Banco de dados: projeto e implementação**. 4. São Paulo Erica 2020.

MILETTO, Evandro M.; BERTAGNOLLI, Silvia C. **Desenvolvimento de software ii: introdução ao desenvolvimento web com html, css, javascript e php**. Porto Alegre Bookman 2014.

MINISTÉRIO DO ESPORTE. **A prática de esporte no Brasil**. 2015. Disponível em: <<http://arquivo.esporte.gov.br/diesporte/2.html>>. Acesso em: 17 jun. 2023.

MORAIS, Izabelly Soares de. **Engenharia de software**. Porto Alegre SAGAH 2020

OLIVEIRA, Cláudio L. V.; ZANETTI, Humberto A. P. **JavaScript descomplicado: programação para a Web, IoT e dispositivos móveis**. São Paulo Erica 2020.

OLIVEIRA, Cláudio L. V.; ZANETTI, Humberto A. P. **Node.js: programe de forma rápida e prática**. São Paulo Expressa 2021.

OLIVEIRA, Ricardo. **LGPD** : como evitar as sanções administrativas. São Paulo Expressa 2021

PICHETTI, Roni F. V.; CORTES, Edinilson S.; PAIXÃO, Vanessa S. M. **Banco de dados**. Porto Alegre SAGAH, 2021.

PRESSMAN, Roger, S.; MAXIM, Bruce R. **Engenharia de software: uma abordagem profissional**. 9. ed. Porto Alegre: AMGH, 2021.

REINEHR, Sheila. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020.

SEBESTA, Robert W. **Conceitos de linguagens de programação**. 11. Porto Alegre Bookman 2018.

SCHACH, Stephen R. **Engenharia de software**. 7. Porto Alegre: ArtMed, 2010.

SOARES, Wallace. **Php 5: conceitos, programação e integração com banco de dados**. 7. São Paulo Erica 2013.

VETORAZZO, Adriana de Souza. **Engenharia de software**. Porto Alegre SAGAH 2018.

WHATSAPP. **Sobre o WhatsApp**. 2023. Disponível em: <<https://www.whatsapp.com/about>>. Acesso em: 29 ago. 2023.