

**CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE FRANCA
“Dr. THOMAZ NOVELINO”**

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

**FRANSÉRGIO DE PAULA MORATO FILHO
LUAN CAMPOS KUHLMANN**

**DESENVOLVIMENTO DE UM SISTEMA DE *SCOUT* DIGITAL PARA PARTIDAS DE
VÔLEI**

Trabalho de graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Carlos Alberto Lucas

FRANCA/SP

2025

DESENVOLVIMENTO DE UM SISTEMA DE SCOUT DIGITAL PARA PARTIDAS DE VÔLEI

FRANSÉRGIO DE PAULA MORATO FILHO¹

LUAN CAMPOS KUHLMANN²

RESUMO

A questão problema deste estudo acadêmico é um processo que ocorre com frequência no segmento esportivo, especificamente em partidas de voleibol. Após consultar alguns amigos, que são técnicos e responsáveis por equipes desta modalidade, identificamos que há uma grande quantidade de informações em uma partida de vôlei que precisam de uma compilação de dados que representem a atuação de cada atleta. Este projeto apresenta uma interface destinada ao gerenciamento de cada atleta, visando otimizar a coleta de dados estatísticos e melhorar a análise de desempenho e performance individual. A proposta possui validações no cadastro de equipes, técnicos e atletas, assegurando a precisão e integridade das informações. Essas validações visam prevenir erros e omissões, melhorando a qualidade dos dados e diminuindo o retrabalho. A interface é amigável e proporciona *feedback* em tempo real aos técnicos de equipes e envolvidos, permitindo correções imediatas e impossibilitando a submissão com dados ausentes. O objetivo foi alcançado com a aplicação do levantamento e análise de requisitos, diagramas, prototipação e validação conceitual.

Palavras-chave: Atletas. Dados. Resultados. Técnicos. Voleibol.

ABSTRACT

The problem question of this academic study is a process that occurs frequently in the sports segment, specifically in volleyball matches. After consulting some friends, who are coaches and responsible for teams in this modality, we identified that there is a large amount of information in a volleyball match that requires a compilation of data that represents the performance of each athlete. This project presents an interface designed for the management of each athlete, aiming to optimize the collection of statistical data and improve the analysis of individual performance and performance. The proposal has validations in the registration of teams, coaches and athletes, ensuring the accuracy and integrity of the information. These validations aim to prevent errors and omissions, improving the quality of the data and reducing rework. The interface is user-friendly and provides real-time feedback to team coaches and those involved, allowing immediate corrections and

¹Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr. Thomaz Novelino – Franca/SP. Endereço eletrônico: fransergio.morato@fatec.sp.gov.br

² Graduação em Análise e Desenvolvimento de Sistemas pela Fatec Dr. Thomaz Novelino – Franca/SP. Endereço eletrônico: luan.kuhkmann@fatec.sp.gov.br

preventing submission with missing data. The objective was achieved with the application of the survey and analysis of requirements, diagrams, prototyping and conceptual validation.

Keywords: Athletes. Coaches. Data. Results. Volleyball.

1-Introdução

Este trabalho estuda o desenvolvimento de um aplicativo móvel para contabilização do *scout* em jogos de vôlei, visando otimizar a coleta de dados estatísticos e melhorar a análise de desempenho.

A questão central consiste em conceber uma solução que simplifique a coleta de dados estatísticos durante as partidas.

As hipóteses sugerem que um aplicativo intuitivo pode melhorar a precisão e eficiência da coleta de dados.

O objetivo principal deste trabalho é elaborar um aplicativo, enquanto os objetivos específicos incluem a identificação das necessidades dos usuários e o *design* preliminar da interface.

A justificativa reside na importância de métodos mais ágeis e precisos para análise de desempenho no esporte.

A metodologia proposta envolve pesquisa de mercado, análise de requisitos, prototipação e validação conceitual.

1.1 Termo da Abertura do Projeto (TAP)

Este documento foi verificado e validado, representando uma tarefa crucial para o gerente de integração. Nele, serão apresentados os artefatos delineados na Estrutura Analítica do Projeto (EAP), fornecendo uma visão geral do projeto.

1.1.1 Termo de Abertura do Projeto

O Termo de Abertura do Projeto (TAP) é um documento essencial que define o início oficial do projeto, delineando seus objetivos, escopo e entregáveis. Sua importância reside na criação de uma base sólida para o gerenciamento do projeto, conforme preconizado pelo PMBOK.

1.1.2 Situação Atual

Atualmente, os técnicos de voleibol amador enfrentam desafios significativos ao realizar o *scout* de pontuação dos atletas. A utilização exclusiva de modelos impressos dificulta a eficiência e precisão do processo, além de aumentar a probabilidade de erros.

1.1.3 Justificativa do Projeto

As fraquezas identificadas na análise SWOT, como a dependência de modelo impressos, destacam a necessidade de uma solução tecnológica. Essa solução visa proporcionar uma experiência mais eficiente e precisa aos técnicos de voleibol amador, melhorando assim a qualidade do *scouting* de pontuação.

1.1.4 Propósito do Projeto

O propósito do projeto é desenvolver uma aplicação que permita aos técnicos de voleibol amador realizarem o *scouting* de pontuação dos atletas de forma eficiente e precisa.

O objetivo é otimizar o tempo de resposta dos técnicos, fornecer controle e relatórios detalhados, valorizando a importância da solução sistêmica.

Para alcançar esse propósito, serão utilizados os princípios do 5W2H: O que: Desenvolver uma aplicação de *scout* de pontuação. Por quê: Para facilitar e melhorar a precisão do processo de *scouting*. Como: Implementando um vídeo tutorial de uso e uma interface intuitiva na aplicação.

1.1.5 Descrição do Projeto

A Estrutura Analítica do Projeto (EAP) será o foco desta seção, detalhando as entregas e atividades necessárias para a conclusão bem-sucedida do projeto. Além disso, serão definidos os passos para a implementação de funcionalidades específicas, como a criação de um CRUD e a aplicação dos conhecimentos adquiridos nos 5º e 6º semestres do curso de ADS.

Algumas funcionalidades chave incluem: Registro e pontuação de jogadores; Geração de relatórios de desempenho; Interface intuitiva para facilitar a utilização.

As etapas do projeto incluirão: Planejamento e definição de requisitos; Desenvolvimento da aplicação; Testes e validação; Implantação e suporte contínuo.

1.1.6 Premissas

As premissas do projeto serão documentadas em um arquivo no Teams, abordando questões como a utilização de ferramentas específicas para gestão do projeto (Miro), modelagem de processos (Camunda) e documentação (Excel).

A solução será orientada para os princípios de *Design-Thinking*, Metodologias Ágeis e usabilidade (IHC) combinada com Bancos de Dados para garantir a eficácia e qualidade do produto.

1.1.7 Restrições

A equipe é composta por 2 membros, que trabalham 8 horas por dia e cursam presencial noturno o curso de ADS, limitando o tempo disponível para execução das atividades do projeto.

Possíveis dificuldades incluem capacitação técnica, alfabetização digital e disponibilidade de recursos humanos e materiais.

1.1.8 Stakeholders

Os principais stakeholders incluem os técnicos de voleibol amador (usuários finais), equipe de desenvolvimento, equipe de *UX Design*, empresa contratante e outros envolvidos no ciclo de vida do projeto.

1.1.9 Riscos

Segurança: Vazamento de dados pessoais dos usuários. Qualidade: Falhas na aplicação que comprometam a precisão do *scouting*. Cronograma: Atrasos no desenvolvimento devido a problemas técnicos ou falta de recursos.

1.1.10 Marco [Cronograma]

O cronograma será detalhado, especificando as atividades e entregas planejadas para cada fase do projeto, desde o desenvolvimento inicial até a conclusão e implantação. Atualizações regulares serão realizadas para garantir o cumprimento dos prazos estabelecidos.

1.1.11 Responsabilidades

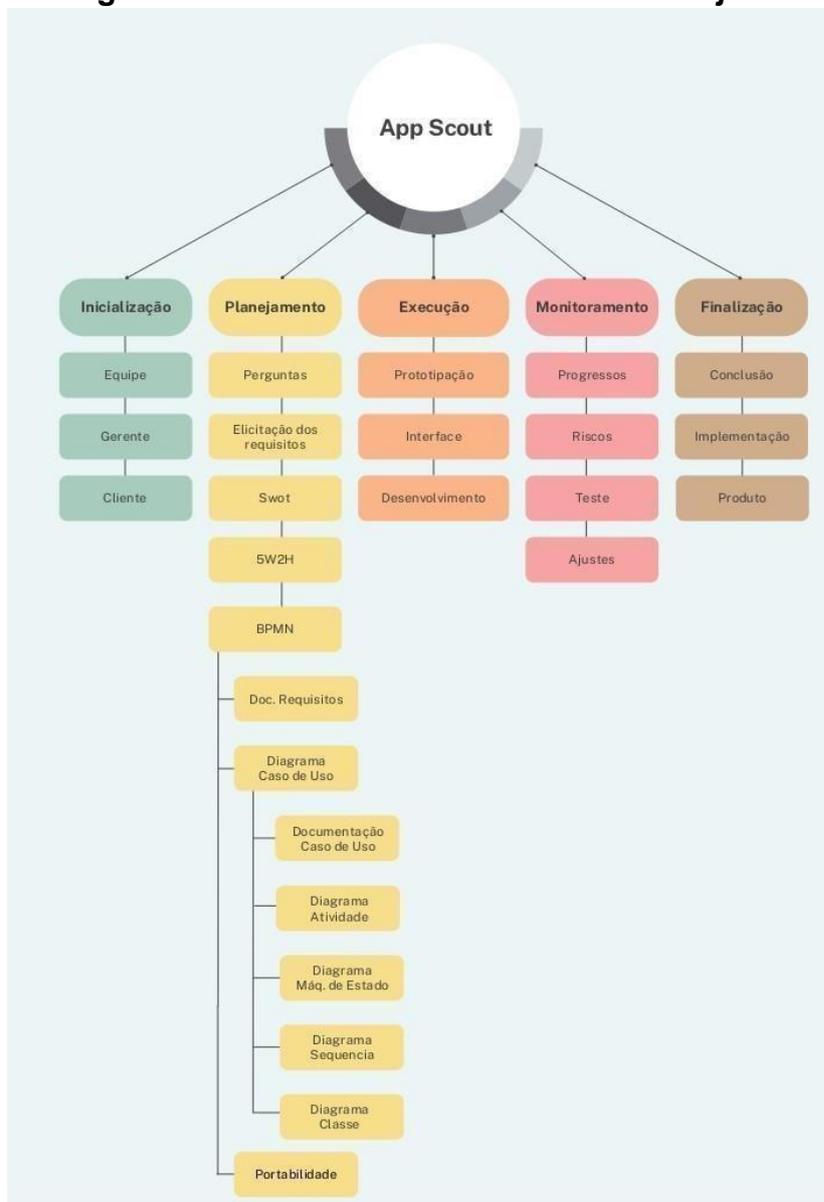
Franségio de Paula Morato Filho

Luan Campos Kuhlmann

1.2 – Estrutura Analítica do Projeto [EAP]

A Estrutura Analítica do Projeto (EAP), representada na figura 1, é utilizada na organização do escopo de um projeto e a hierarquia das atividades a serem concluídas.

Figura 1 - EAP – Estrutura Analítica do Projeto



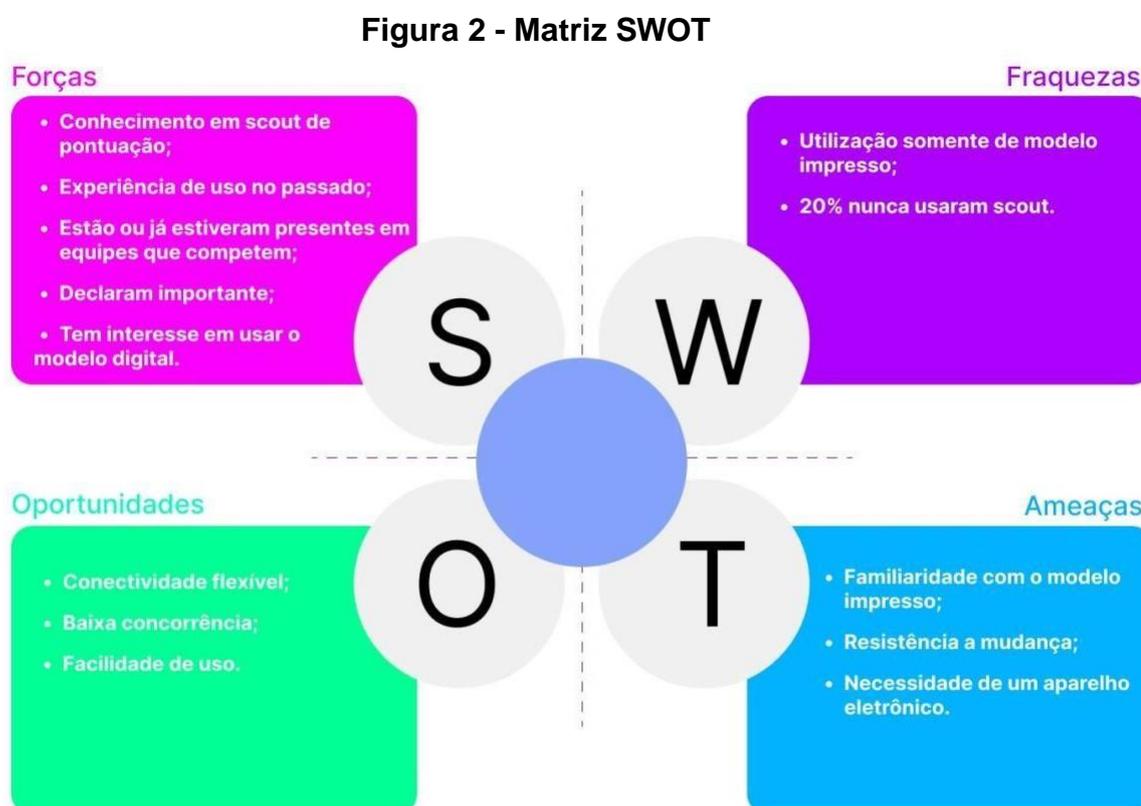
Fonte: autoria própria

2-Viabilidade do Projeto

2.1 Matriz SWOT

Análise SWOT é uma técnica de planejamento estratégico utilizada para auxiliar pessoas ou organizações a identificar forças, fraquezas, oportunidades, e ameaças relacionadas à competição em negócios ou planejamento de projetos.

Esta técnica tem como objetivo especificar os riscos do negócio ou projeto, e identificar os fatores internos e externos que são favoráveis e desfavoráveis para alcançar esses objetivos.



Fonte: autoria própria

Os usuários potenciais do aplicativo possuem um sólido conhecimento em *scout* de pontuação, o que facilita a adaptação a um novo modelo digital. A experiência prévia no uso de *scouts*, mesmo que em formatos impressos, proporciona uma base sólida para essa transição tecnológica.

Além disso, muitos desses usuários estão ou estiveram presentes em equipes de vôlei que competem regularmente, o que destaca a importância e a utilidade de um sistema eficiente de *scout*.

Há também um reconhecimento claro da importância do *scout* para a análise de desempenho, o que aumenta a motivação para adotar novas ferramentas tecnológicas.

Outro ponto positivo é o interesse declarado por parte dos usuários em migrar para um modelo digital, reconhecendo os benefícios que a tecnologia pode oferecer. No entanto, uma das principais fraquezas identificadas é a utilização predominante de modelos impressos de *scout*, indicando uma resistência inicial à mudança para métodos digitais devido à falta de familiaridade com a tecnologia.

Além disso, cerca de 20% dos potenciais usuários nunca utilizaram *scout*, o que representa um desafio na educação e adaptação a um novo sistema digital. Existem, contudo, oportunidades significativas a serem exploradas.

A conectividade flexível do aplicativo, que permite sua integração com diferentes dispositivos e plataformas, oferece uma flexibilidade que pode facilitar a adoção do novo sistema em diversos contextos de uso. A baixa concorrência no mercado de aplicativos para *scout* de vôlei cria uma oportunidade para se estabelecer como líder e pioneiro nesse campo.

Além disso, a facilidade de uso do aplicativo, com uma interface intuitiva e amigável, pode reduzir a resistência inicial e promover uma adoção mais rápida entre os usuários. Apesar das oportunidades, há ameaças consideráveis a serem enfrentadas.

A familiaridade de muitos usuários com o modelo impresso pode representar uma barreira significativa à adoção de novos métodos digitais, devido ao conforto e à tradição associada aos processos antigos.

A resistência inerente a mudanças tecnológicas, especialmente entre usuários mais acostumados a métodos antigos, pode dificultar a transição para o novo sistema.

Finalmente, a necessidade de um dispositivo eletrônico para utilizar o aplicativo pode ser um impedimento para alguns usuários, seja por questões de acesso, custo ou preferência pessoal.

2.2 Plano de Ação - 5W2H

A matriz 5W2H é uma ferramenta de gestão utilizada para planejamento e análise de processos, projetos ou problemas.

Ela é composta por sete perguntas chave que ajudam a esclarecer e detalhar aspectos importantes do que está sendo analisado.

As perguntas são: What (O que?), Why (Por que?), Who (Quem?), Where (Onde?), When (Quando?), How (Como?) e How much (Quanto custa?).

Cada uma dessas perguntas oferece uma perspectiva específica e complementa as demais, proporcionando uma visão abrangente e detalhada.

Figura 3 - Matiz 5W2H

| Questão problema - 1 = <u>Utilização somente de modelo impresso</u> | | | | | | | |
|---|---|----------------------------------|--------------------|---------------------------|---------------------------------------|--|---------------------|
| Ação | O que? (What?) | Por que? (Why?) | Onde? (Where?) | Quem? (Who?) | Quando? (When?) | Como? (How?) | Quanto? (How much?) |
| 1 | Inserir vídeo de tutorial de uso dentro da aplicação. | Para facilitar o uso do usuário. | Na aplicação final | Equipe de desenvolvimento | Média de 1 a 2 anos para implantação | <ul style="list-style-type: none"> - Produzindo o vídeo - Indexando o vídeo dentro da aplicação - Adicionando reprodutor de vídeo - Testando se funcionou de forma adequada - Aplicando solução | Resultados em Junho |
| 2 | Produzir uma interface intuitiva. | Para facilitar o uso do usuário. | Na aplicação final | Equipe de UX Design. | Média de 1 a 2 anos para implantação. | <ul style="list-style-type: none"> - Produzindo a interface - Implementando interface em código - Realizando testes - Aplicando solução | Resultados em Junho |

Fonte: autoria própria

3-Levantamento de Requisitos

3.1 Elicitação e especificação dos Requisitos

As técnicas de elicitação de requisitos são métodos essenciais para identificar e coletar as necessidades e expectativas dos stakeholders em relação ao sistema a ser desenvolvido.

Este processo pode envolver diversas abordagens, como entrevistas, questionários, *workshops*, observação direta e análise de documentos existentes. Cada técnica contribui para uma compreensão abrangente das demandas e dos problemas que o sistema precisa resolver.

Após a elicitação, segue-se a especificação dos requisitos, que consiste em documentar de maneira clara, precisa e detalhada as informações coletadas. Esta

documentação serve como base para o desenvolvimento, garantindo um entendimento comum e claro entre todos os envolvidos sobre o que será desenvolvido.

Neste projeto, iniciamos o processo de elicitação de requisitos com um formulário eletrônico detalhado, direcionado a treinadores e técnicos de times de vôlei de Franca-SP.

A aplicação do questionário foi crucial para o projeto, pois permitiu identificar as necessidades específicas dos usuários e orientar o desenvolvimento do sistema de forma precisa e alinhada às expectativas dos *stakeholders*.

Além disso, realizamos uma análise SWOT para identificar as forças, fraquezas, oportunidades e ameaças relacionadas ao negócio. Durante essa análise, identificamos que uma das principais fraquezas era a utilização predominante de modelos impressos de *scout*, indicando uma resistência inicial à mudança para métodos digitais.

A familiaridade e o conforto com o formato tradicional podem representar uma barreira significativa à adoção do novo sistema. No entanto, essa análise também destacou várias oportunidades, como a baixa concorrência no mercado de aplicativos para *scout* de vôlei e a flexibilidade de conectividade que a solução digital oferece.

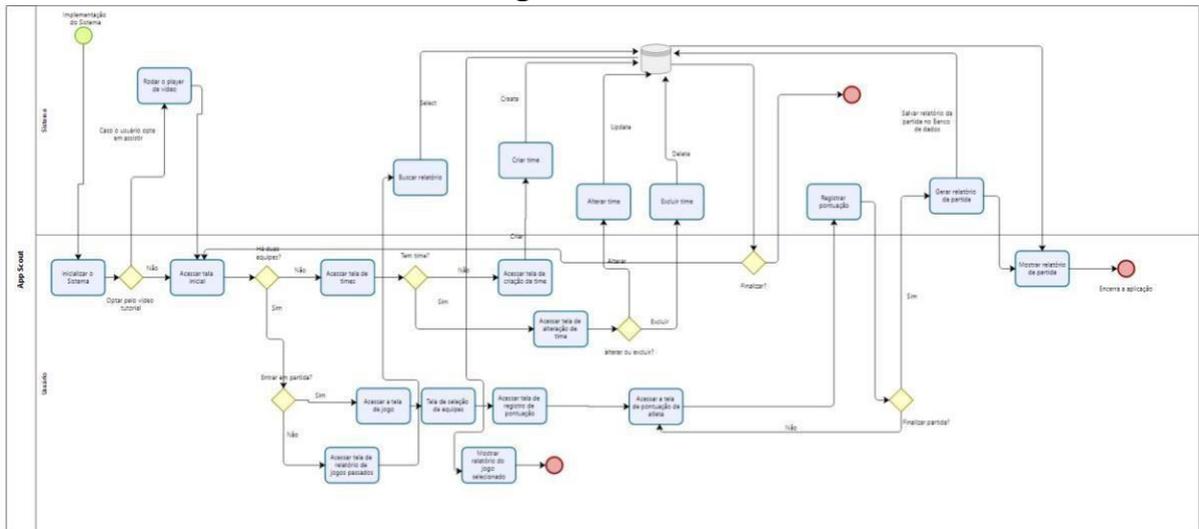
Em resumo, a combinação da elicitação de requisitos detalhada e da análise SWOT em conjunto do uso da ferramenta 5W2H nos forneceu uma visão clara dos desafios e das oportunidades associadas ao desenvolvimento do aplicativo para o início da modelagem do BPMN.

3.2 BPMN

BPMN (Business Process Model and Notation) é uma linguagem gráfica padronizada para representar processos de negócio de forma clara e concisa. Através de símbolos universais, ele facilita a comunicação e colaboração entre diferentes áreas de uma organização, desde analistas de negócios até desenvolvedores de software.

Com o BPMN, é possível visualizar um processo na íntegra, identificar gargalos, comunicar o processo de forma clara e até mesmo automatizá-lo. Isso resulta em diversos benefícios, como aumento da eficiência e produtividade, redução de custos, maior agilidade e melhorando o processo de tomada de decisão (BPMN.IO, 2024).

Figura 4 - BPMN



Fonte: autoria própria

O BPMN foi modelado com duas raias, representando o sistema e o usuário. A implementação do sistema inicia-se quando o usuário entra no aplicativo, onde ele pode optar por assistir ao vídeo tutorial ou pular essa etapa. Após a conclusão do vídeo ou a escolha de não o assistir, o usuário é direcionado para a tela principal, que apresenta diversas opções de interação.

Na tela principal, o usuário pode iniciar uma nova partida ou acessar uma partida existente, definindo o nome do jogo e os times que competirão. Em seguida, ele é direcionado para a tela de marcação de pontos. Além disso, o usuário tem a opção de acessar a tela de gerenciamento de times, onde pode criar times, alterar informações ou deletar times já existentes.

Após o registro da pontuação e o término do jogo, o usuário salva a partida e tem acesso ao relatório. Este relatório pode ser detalhado por time ou por jogador, oferecendo uma visão abrangente das estatísticas da partida.

3.3 Requisitos Funcionais

Na engenharia de software, os Requisitos Funcionais (RFs) são a base para o desenvolvimento de um sistema que atenda às necessidades e expectativas dos usuários. É por meio deles que é definido de forma clara e objetiva o que o sistema deverá fazer.

Ou seja, os RFs são uma espécie de manual de instrução para os desenvolve-

dores, detalhando as funcionalidades que o sistema deverá oferecer. Cada RF descreve uma tarefa específica que o sistema precisa realizar. Os requisitos funcionais identificados no sistema de gerenciamento de professores orientadores foram:

Quadro 1 – Requisitos Funcionais do sistema

| | |
|---------------------------------|---|
| RF001 | Optar pelo vídeo tutorial. |
| Descrição → | O usuário deverá escolher se deseja ou não assistir o vídeo tutorial. |
| Categoria: evidente | Prioridades: desejável |
| Informações → | Para a implementação desse requisito será necessário aparecer uma janela no sistema sempre que for iniciado, |
| | perguntando ao usuário se ele deseja ou não assistir o vídeo de tutorial. Deverá também ter uma caixinha com a opção de “Não perguntar mais”. |
| Regra de Negócio (se existir) → | Não há regra. |
| RF002 | Rodar o player de vídeo. |
| Descrição → | O sistema deverá reproduzir o vídeo de tutorial do sistema. |
| Categoria: evidente | Prioridades: obrigatório |
| Informações → | Para a implementação desse requisito será necessário que o sistema reproduza o vídeo de tutorial que estará inserido dentro da aplicação. |
| Regra de Negócio (se existir) → | Não há regra. |
| RF003 | Acessar a tela inicial. |
| Descrição → | O usuário deverá acessar a tela inicial do sistema. |
| Categoria: evidente | Prioridades: obrigatório |
| Informações → | Para a implementação desse requisito será necessário que o sistema apresente ao usuário a tela de tutorial. |
| Regra de Negócio (se existir) → | Não há regra. |
| RF004 | Acessar a tela de times. |
| Descrição → | O usuário deverá acessar a tela de times. |
| Categoria: evidente | Prioridades: obrigatório |
| Informações → | Para a implementação desse requisito será necessário que o sistema faça com que o usuário acesse a tela de cadastro de equipe. |
| Regra de Negócio (se | Não há regra. |

| | |
|---------------------------------|--|
| existir) → | |
| RF005 | Acessar tela de criação de time. |
| Descrição → | O usuário deverá acessar a tela de criação de times. |
| Categoria: evidente | Prioridades: obrigatório |
| | <p>Para a implementação desse requisito será necessário criar uma tela de cadastro com “Nome do time” em cima. Abaixo deverá haver os seguintes campos: “Nome do jogador”, “Posição do jogador”, “Altura” e “Braço forte” (destro ou canhoto) e o botão de “Incluir jogador” e o botão de “Cancelar”.</p> <p>Além disso será necessário incluir abaixo um botão de “Finalizar” e “Cancelar”.</p> |
| Regra de Negócio (se existir) → | O time não pode haver menos que 6 jogadores e mais que 14 jogadores cadastrados. |
| RF006 | Criar time. |
| Descrição → | O sistema deverá armazenar no banco de dados o time cadastrado. |
| Categoria: oculto | Prioridades: obrigatório |
| Informações → | Para a implementação desse requisito será necessário que o sistema |
| | armazene as informações cadastradas no banco de dados. |
| Regra de Negócio (se existir) → | Não há regra. |
| RF007 | Acessar tela de alteração de time. |
| Descrição → | O usuário deverá acessar a tela de alteração de times. |
| Categoria: evidente | Prioridades: obrigatório |
| Informações → | <p>Para a implementação desse requisito será necessário criar uma tela de cadastro com “Nome do time” em cima. Abaixo deverá haver os seguintes campos: “Nome do jogador”, “Posição do jogador”, “Altura” e “Braço forte” (destro ou canhoto) e o botão de “Alterar jogador” e o botão de “Cancelar”.</p> <p>Além disso será necessário incluir abaixo um botão de “Finalizar” e “Cancelar”</p> |

| | |
|---------------------------------|--|
| Regra de Negócio (se existir) → | O time não pode haver menos que 6 jogadores e mais que 14 jogadores cadastrados. |
| RF008 | Alterar time. |
| Descrição → | O sistema deverá alterar os dados do time. |
| Categoria: oculto | Prioridades: obrigatório |
| Informações → | Para a implementação desse requisito será necessário que o sistema faça um update no banco de dados das informações do time inseridas pelo usuário. |
| Regra de Negócio (se existir) → | Não há regra. |
| RF009 | Deletar time. |
| Descrição → | O sistema deverá apagar os dados do time. |
| Categoria: oculto | Prioridades: obrigatório |
| Informações → | Para a implementação desse requisito será necessário que o sistema apague o time. |
| Regra de Negócio (se existir) → | Não há regra. |
| RF010 | Acessar tela de jogo. |
| Descrição → | O usuário deverá acessar a tela de jogo. |
| Categoria: evidente | Prioridades: obrigatório |
| Informações → | Para a implementação desse requisito será necessário criar uma tela com os botões “Iniciar partida”, “Cancelar”. Além disso o jogo deverá dar um nome a partida. Exemplo “Partida 1”. |
| Regra de Negócio (se existir) → | Não há regra. |
| RF011 | Tela de seleção de equipes. |
| Descrição → | O usuário deverá ser apresentado os times que já estão cadastrados e escolher 2 deles para o jogo. |
| Categoria: evidente | Prioridades: obrigatório |
| Informações → | Para a implementação desse requisito será necessário criar uma tela apresentando os times cadastrados. Permitindo que usuário selecione até 2 times para iniciar a partida. Além de quatro botões, sendo eles: “Inserir time”, “Remover time”, “Pronto” e o botão de “Cancelar”. |
| Regra de Negócio (se existir) → | O jogo não pode iniciar sem haver 2 times inseridos na partida |
| RF012 | Acessar tela de registro de pontuação. |

| | |
|---------------------------------|---|
| Descrição → | O usuário deverá acessar a tela de alteração de registro de pontuação. |
| Categoria: evidente | Prioridades: obrigatório |
| Informações → | Para a implementação desse requisito será necessário criar uma tela mostrando os dois times em partida e o placar do jogo. Além disso o sistema deve listar os jogadores de cada equipe com sua pontuação atual ao lado de seu nome. Havendo uma divisória no meio separando os jogadores de times diferentes. |
| Regra de Negócio (se existir) → | Não há regra. |
| RF013 | Acessar tela de registro de pontuação de atleta. |
| Descrição → | O usuário deverá acessar a tela de pontuação de atleta. |
| Categoria: evidente | Prioridades: obrigatório |
| Regra de Negócio (se existir) → | Só será possível aumentar o placar de um time após registrar a pontuação do atleta. Se for positiva deverá incluir um ponto no placar do time que ele está. Caso seja negativa deverá ser incluído um ponto no placar do time inimigo. |
| RF014 | Gerar relatório da partida |
| Descrição → | O sistema deverá gerar o relatório da partida |
| Categoria: oculto | Prioridades: obrigatório |
| Informações → | Para a implementação desse requisito será necessário que o sistema gere o relatório com o placar final do jogo e o ranking de pontuação de cada jogador presente nas duas equipes. Logo após esse relatório deverá ser salvo no banco de dados para que possa ser consultado futuramente. |
| Regra de Negócio (se existir) → | Não há regra. |
| RF015 | Acessar tela de alteração de time. |
| Descrição → | O usuário deverá acessar a tela de alteração de times. |
| Categoria: evidente | Prioridades: obrigatório |
| Informações → | Para a implementação desse requisito será necessário criar uma tela de cadastro com "Nome do time" em cima. Abaixo deverá haver os seguintes campos: "Nome do jogador", "Posição do jogador", |
| | "Altura" e "Braço forte" |

| | |
|---------------------------------|---|
| | (destro ou canhoto) e o botão de “Alterar jogador” e o botão de “Cancelar”. Além disso será necessário incluir abaixo um botão de “Finalizar” e “Cancelar” |
| Regra de Negócio (se existir) → | |
| RF016 | Mostrar relatório da partida. |
| Descrição → | O usuário deverá acessar a tela de relatório da partida. |
| Categoria: evidente | Prioridades: obrigatório |
| Informações → | Para a implementação desse requisito será necessário criar uma tela de apresentando o relatório de partida criado no RF014. Além disso será necessário incluir abaixo um botão de “Finalizar” que deverá fechar a aplicação ao ser clicado. |
| Regra de Negócio (se existir) → | Não há regra. |
| RF017 | Acessar tela de relatório de jogos passados. |
| Descrição → | O usuário deverá acessar a tela com o relatório dos jogos passados. |
| Categoria: evidente | Prioridades: obrigatório |
| Informações → | Para a implementação deste requisito deverá ser apresentado ao usuário a lista de jogos passados. Aparecendo o nome da partida e o placar final do |
| | jogo. Permitindo ao usuário clicar no jogo que deseja ver o relatório passado. Ao clicar na partida selecionada deverá aparecer um alerta perguntando se o usuário deseja acessar o relatório da partida selecionada. Com dois botões, sendo eles: “Confirmar” e “Voltar” |
| Regra de Negócio (se existir) → | Não há regra. |
| RF018 | Buscar relatório. |
| Descrição → | O sistema deverá buscar o relatório da partida selecionada. |
| Categoria: oculto | Prioridades: obrigatório |
| Informações → | Para a implementação desse requisito será necessário que o sistema acesse o banco de dados e busque o relatório da partida selecionada. |
| Regra de Negócio (se existir) → | Não há regra. |

| | |
|---------------------------------|--|
| RF019 | Mostrar relatório do jogo selecionado. |
| Descrição → | O usuário deverá acessar a tela com o relatório do jogo selecionado. |
| Categoria: evidente | Prioridades: obrigatório |
| Informações → | Para a implementação desse requisito será necessário criar uma tela apresentando o relatório do jogo selecionado seguindo o modelo do RF015. |
| Regra de Negócio (se existir) → | Não há regra. |

Fonte: autoria própria

3.4 Regras de Negócio

No âmbito da engenharia de software, uma regra de negócio é uma especificação que define ou restringe aspectos do comportamento do sistema, alinhando-o aos objetivos e políticas da organização (ALURA, 2021).

Essas regras determinam como os dados devem ser criados, armazenados e modificados, além de descreverem as operações permitidas sobre eles, garantindo que o software funcione conforme as necessidades do negócio.

A relevância das regras de negócio está em sua capacidade de assegurar que o software atende aos requisitos específicos da organização, promovendo consistência, precisão e eficiência nas operações (ALURA, 2021).

Elas ajudam a reduzir ambiguidades e erros, fornecendo diretrizes claras para desenvolvedores e partes interessadas, e garantem que todos os componentes do sistema colaboram de maneira coerente para alcançar os objetivos estratégicos da empresa.

Além disso, as regras de negócio facilitam a manutenção e a evolução do sistema, centralizando o conhecimento das operações e políticas da empresa. Isso permite que modificações sejam implementadas de forma mais ágil e com menor risco de impacto negativo, preservando a integridade e a qualidade do software ao longo do tempo.

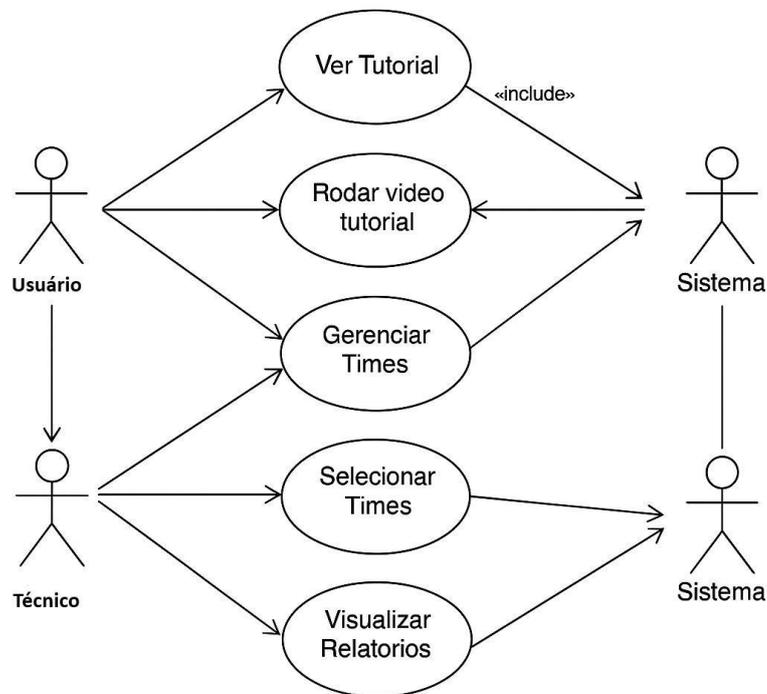
No sistema de *scout* de vôlei, entretanto, não há regras de negócio definidas. Portanto, não será apresentado um quadro de regras de negócio.

3.5 Casos de Uso

Na Engenharia de Software, os Casos de Uso são ferramentas essenciais para modelar e documentar as funcionalidades de um sistema, detalhando como os usuários interagem com ele para executar tarefas específicas. Normalmente, os casos de uso são representados por diagramas que facilitam a visualização das funcionalidades do sistema de maneira clara e acessível (CONCEITO DE, 2021).

Esses casos de uso desempenham um papel crucial na análise e no *design* de sistemas, servindo para documentar requisitos, entender os fluxos de trabalho dos usuários e validar a arquitetura do sistema.

Figura 5 - Diagrama de Caso de Uso



Fonte: autoria própria

3.6 Diagrama de Classes

Um diagrama de classes é uma representação visual usada na engenharia de software para descrever a estrutura de um sistema orientado a objetos. Ele exhibe as classes que compõem o sistema e os relacionamentos entre elas.

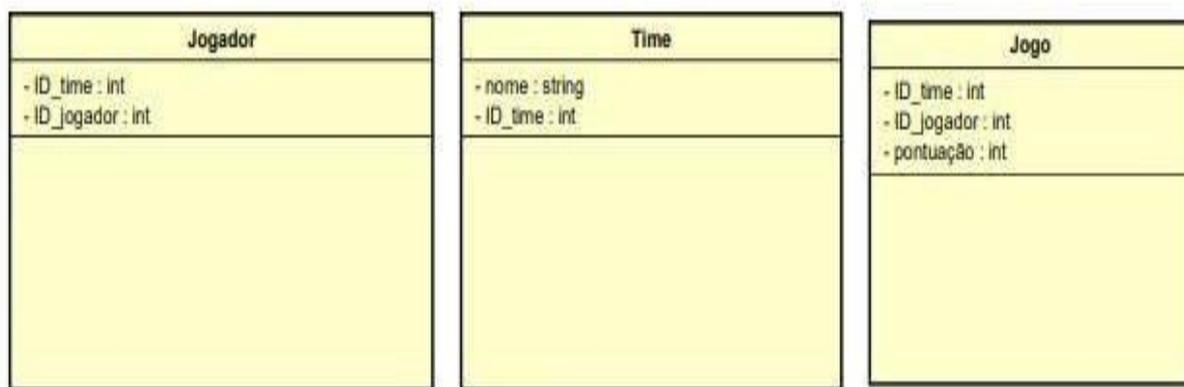
Cada classe é representada por um retângulo dividido em três seções: o nome da classe, seus atributos (ou propriedades) e os métodos (ou operações) que a classe

pode executar. As linhas que conectam os retângulos indicam os diferentes tipos de relacionamentos, como associações, heranças e dependências (PROJECT BUILDER, 2021).

A importância do diagrama de classes na engenharia de software é significativa. Ele fornece uma visão clara e organizada da estrutura do sistema, ajudando a entender como os diferentes componentes interagem e se relacionam.

Essa visão é crucial para o design do sistema, permitindo que os desenvolvedores identifiquem e planejem a implementação das classes e suas interações de maneira coesa e eficiente.

Figura 6 - Diagrama de Classes



Fonte: autoria própria

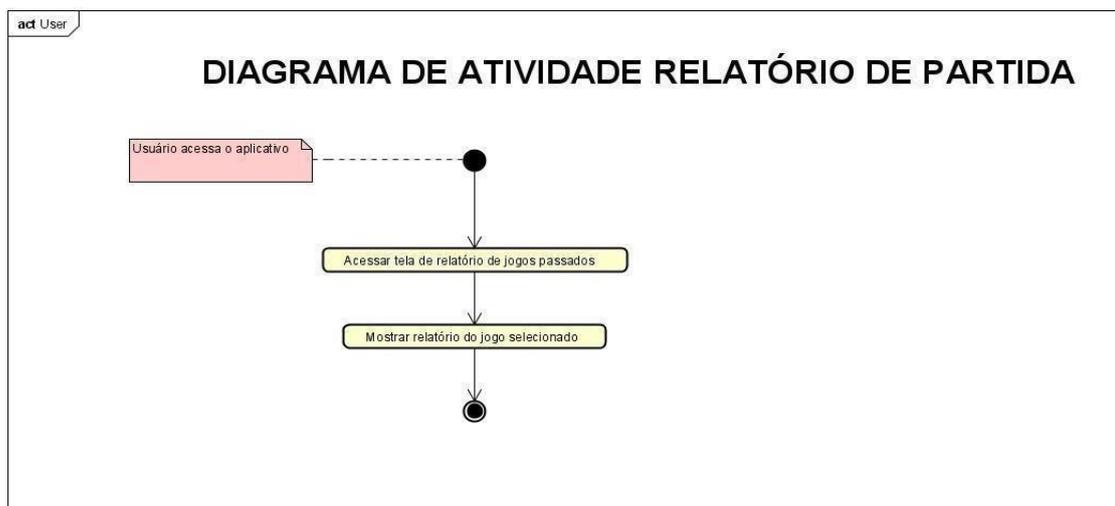
3.7 Diagrama de Atividades

O diagrama de atividades é uma ferramenta visual na engenharia de software usada para modelar o fluxo de trabalho ou os processos dentro de um sistema.

Ele representa a sequência de atividades, decisões e interações que ocorrem em um processo específico, utilizando símbolos padronizados, como elipses para atividades, losangos para decisões e setas para indicar a direção do fluxo.

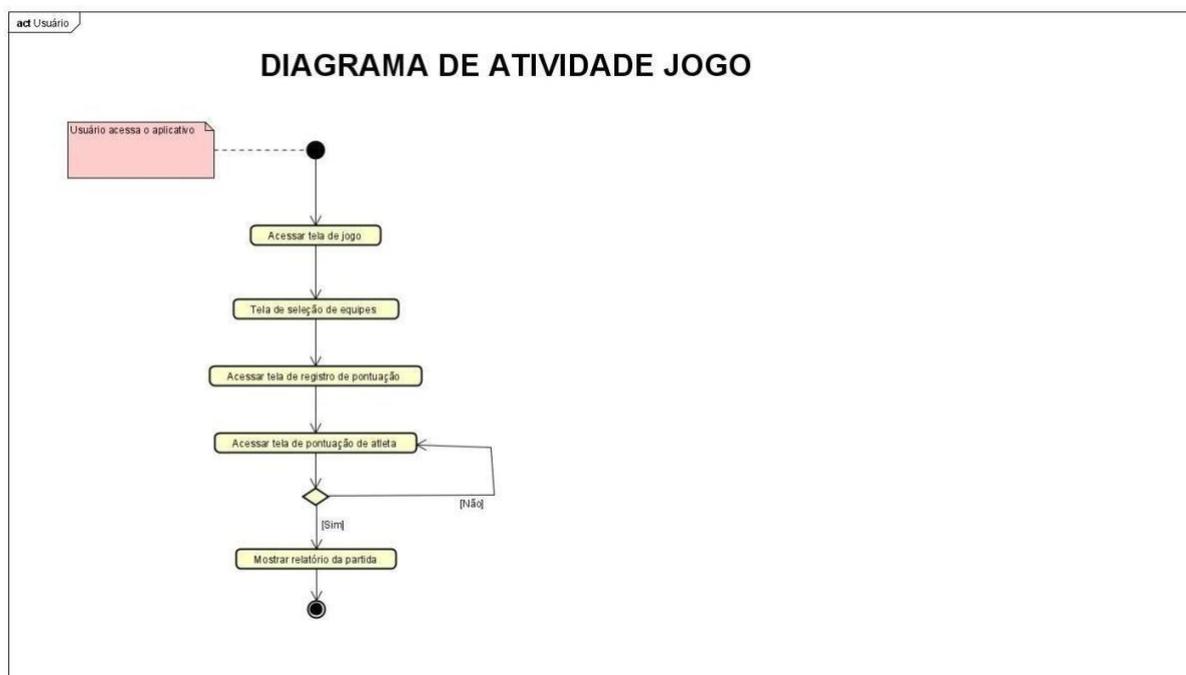
A principal função do diagrama de atividades é ilustrar claramente como diferentes ações se encadeiam e se inter-relacionam, permitindo a identificação de pontos críticos, gargalos e possíveis melhorias no processo.

Figura 7 - Diagrama de Atividade Relatório de Partida



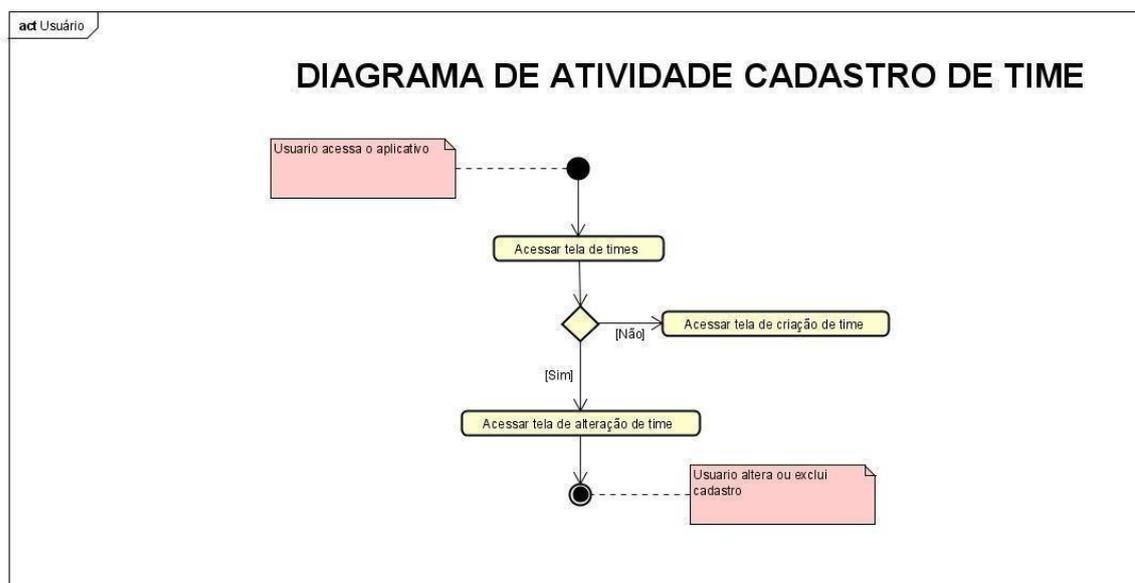
Fonte: autoria própria

Figura 8 - Diagrama de Atividade de Jogo



Fonte: autoria própria

Figura 9 - Diagrama de Atividade Cadastro de Time



Fonte: autoria própria

3.8 Diagrama de Estados

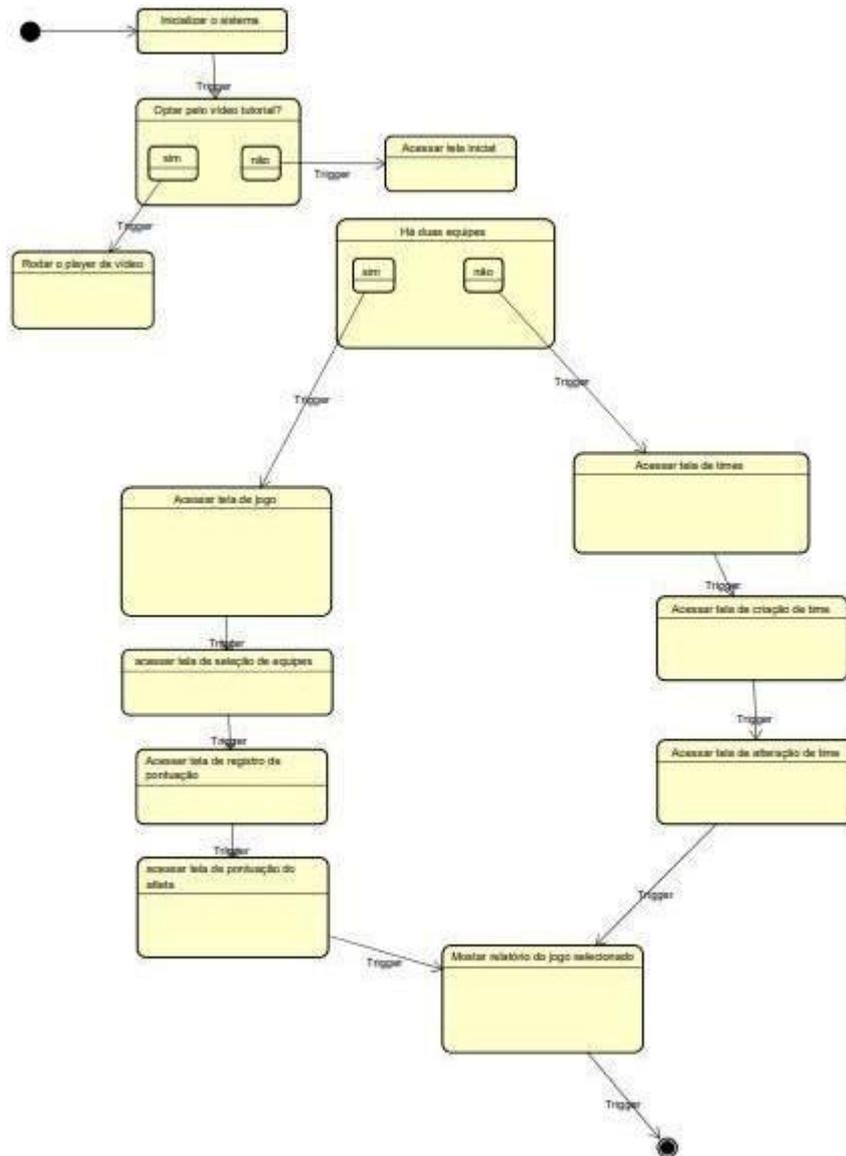
O diagrama de máquina de estados é uma ferramenta de modelagem utilizada na engenharia de software para ilustrar o comportamento de um sistema ou objeto por meio de seus estados e das transições entre eles.

Cada estado representa uma condição específica na vida do objeto, enquanto as transições, desencadeadas por eventos específicos, mostram a mudança de um estado para outro.

A principal função deste diagrama é capturar a lógica de controle do sistema, detalhando como ele deve responder a diferentes eventos conforme seu estado atual.

Os elementos fundamentais do diagrama incluem estados, transições, eventos e ações, além dos estados iniciais e finais que marcam o início e o fim do ciclo de vida do objeto.

Figura 10 - Diagrama de Máquina de Estado



Fonte: autoria própria

3.9 Diagrama de Sequência

O diagrama de sequência é uma ferramenta crucial na engenharia de software, amplamente utilizada para modelar a interação entre objetos dentro de um sistema ao longo do tempo.

Como parte da UML (*Unified Modeling Language*), este diagrama é particularmente eficaz para ilustrar a ordem e o fluxo das mensagens trocadas entre os componentes do sistema. Ele é valioso para compreender como diferentes partes de um software colaboram para realizar funções específicas, oferecendo uma visão detalhada do comportamento dinâmico do sistema.

No diagrama de sequência, cada objeto participante é representado por uma linha de vida, que é uma linha vertical indicando a existência do objeto ao longo do tempo.

As interações entre esses objetos são mostradas por setas horizontais que cruzam as linhas de vida, exibindo a direção e a sequência das mensagens ou chamadas de métodos.

Essas setas são etiquetadas com os nomes das mensagens e, frequentemente, com parâmetros e retornos de valores, tornando explícita a comunicação entre os objetos. Além disso, ativações, representadas por barras retangulares ao longo das linhas de vida, indicam a duração da execução de operações ou métodos específicos por esses objetos.

A principal função do diagrama de sequência é detalhar a lógica de execução dos processos do sistema, oferecendo uma visão clara e organizada das interações entre os objetos.

Ele permite que os desenvolvedores compreendam melhor como as funções são realizadas, identifiquem possíveis pontos de falha ou ineficiências e otimizem o fluxo de trabalho.

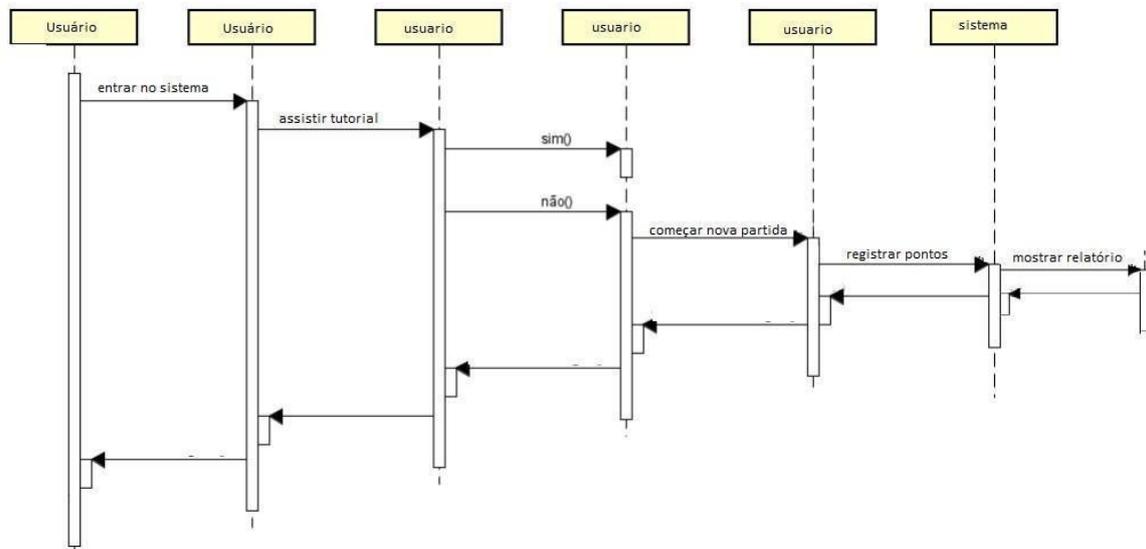
Esse diagrama é particularmente útil durante a fase de design do software, pois ajuda a definir e verificar a lógica dos processos antes da implementação, garantindo que todos os requisitos funcionais sejam atendidos de maneira eficaz.

Em resumo, o diagrama de sequência é uma ferramenta essencial para a análise e modelagem do comportamento dinâmico de sistemas na engenharia de software.

Ele facilita a comunicação entre desenvolvedores, analistas e outras partes interessadas, proporcionando uma representação visual clara e detalhada das interações entre objetos.

Ao destacar a sequência temporal das operações, o diagrama de sequência contribui significativamente para a construção de sistemas robustos e eficientes, assegurando que todos os componentes funcionem de maneira coordenada para alcançar os objetivos do sistema.

Figura 11 - Diagrama de Sequencia



Fonte: autoria própria

3.10 Matriz de Rastreabilidade

Uma matriz de rastreabilidade é uma ferramenta na engenharia de software usada para mapear e rastrear a relação entre diferentes elementos de um projeto. Esses elementos podem incluir requisitos, casos de uso, testes, módulos de código, entre outros.

A matriz geralmente é apresentada como uma tabela, onde os itens de um conjunto são listados em uma dimensão e os itens relacionados de outro conjunto são listados na outra dimensão, permitindo que as interações e dependências entre eles sejam visualizadas facilmente.

Em primeiro lugar, ela assegura que todos os requisitos especificados inicialmente sejam devidamente implementados e validados. Ao mapear cada requisito aos componentes correspondentes do sistema, como casos de uso e testes, a matriz ajuda a garantir que nenhuma necessidade dos *stakeholders* seja negligenciada durante o desenvolvimento.

Além disso, a matriz de rastreabilidade é essencial para a gestão de mudanças. Quando um requisito é alterado, a matriz permite identificar rapidamente quais partes do sistema serão afetadas, facilitando a análise de impacto e a implementação de alterações de forma controlada. Isso ajuda a evitar regressões e a garantir que o sistema se mantenha consistente e funcional após as modificações.

Quadro 2 – Matriz de Rastreabilidade

| X | RN01 | RN02 | RN03 | RN04 |
|-------|------|------|------|------|
| RF001 | | | | |
| RF002 | | | | |
| RF003 | | | | |
| RF004 | | | | |
| RF005 | X | | | |
| RF006 | | | | |
| RF007 | | X | | |
| RF008 | | | | |
| RF009 | | | | |
| RF010 | | | | |
| RF011 | | | X | |
| RF012 | | | | |
| RF013 | | | | X |
| RF014 | | | | |
| RF015 | | | | |
| RF016 | | | | |
| RF017 | | | | |
| RF018 | | | | |
| RF019 | | | | |

Fonte: autoria própria

4-Ferramentas e Métodos

As ferramentas escolhidas para o projeto foram selecionadas com base em sua eficiência, escalabilidade e suporte à comunidade.

Além disso, essas ferramentas têm documentação abrangente, tutoriais e recursos disponíveis na comunidade de desenvolvedores, o que torna mais fácil para os desenvolvedores aprenderem e implementarem as soluções.

A escolha dessas ferramentas também foi influenciada pela preferência pessoal dos autores deste trabalho e experiência prévia no uso delas.

A maioria possui licença de código aberto, o que significa que são gratuitas e podem ser usadas para fins comerciais e pessoais. São elas:

SWOT: Figma

Versão: Educacional

5W2H: Word

Versão: Educacional [365]

BPMN: Bizagi

Versão: Educacional [v10]

DOCUMENTAÇÃO: Word

Versão: Educacional [365]

DIAGRAMAS: Astah

Versão: 10.1

FRONT: Vue.js

Versão: 3.2

BACK: Java (REDUX, 2024)

Versão: SE 24

BANCO DE DADOS: PostgreSQL

Versão: 17.5

ROTAS: SpringBoot

Versão: 3.5.1

5-Desenvolvimento

A tela principal do aplicativo possui 3 sessões, sendo elas, Times, Jogador e Técnicos. Haverá também a tela de cadastro de jogo.

Figura 12 – Tela principal

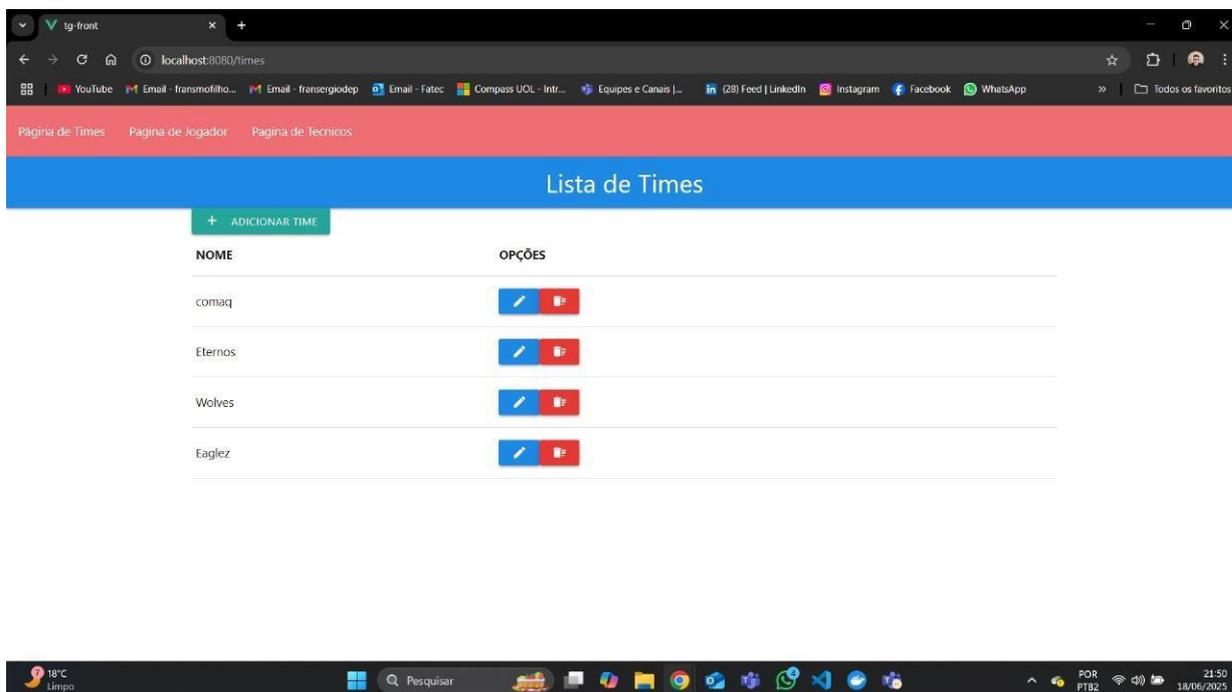


Fonte: autoria própria

Ao entrar na sessão de times o usuário tem acesso a todas as equipes que já foram cadastradas. Nelas possuem a opção de editá-las ou deletá-las. Além disso,

possui um botão para cadastro de novas equipes.

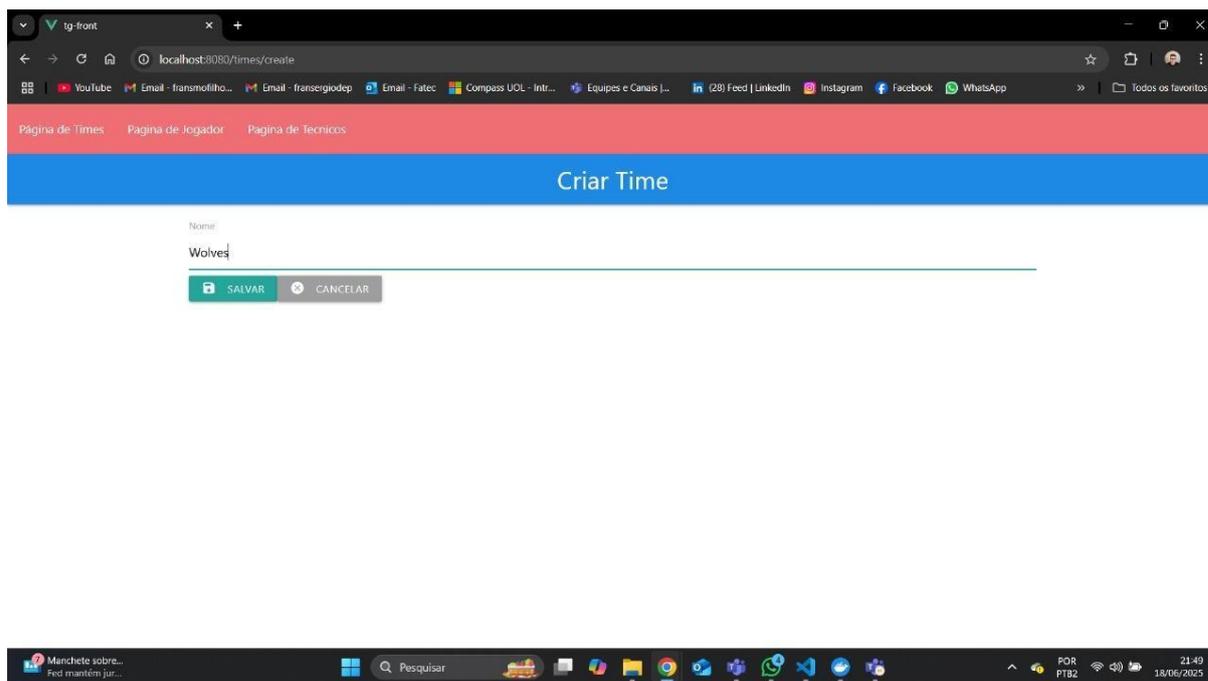
Figura 13 – Tela de equipes



Fonte: autoria própria

Na tela de cadastro que equipes o usuário deve preencher o nome da equipe que deseja incluir. Após isso deve clicar em salvar para registrar.

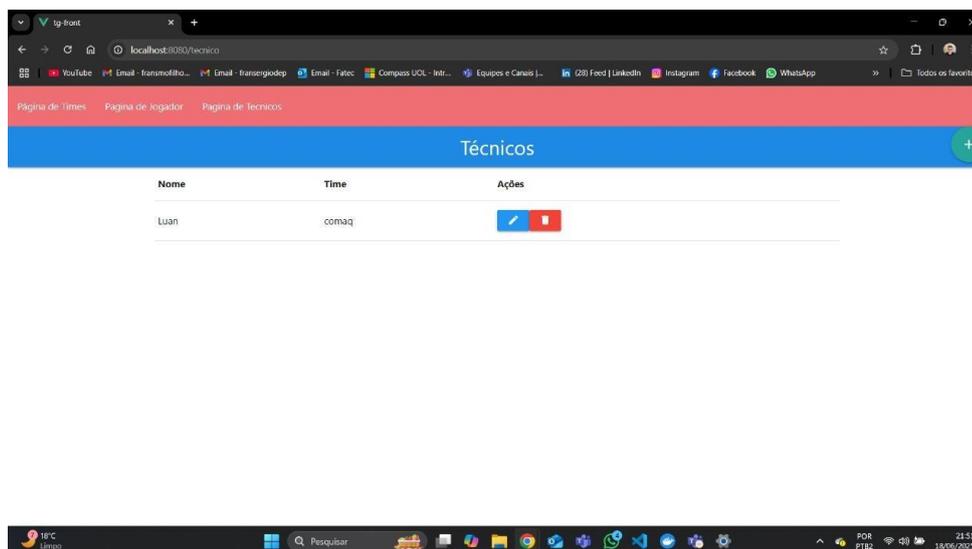
Figura 14 – Tela de cadastro de equipes



Fonte: autoria própria

Ao clicar na aba de Técnicos o usuário terá acesso a tela que mostra todos os técnicos cadastrados e as equipes pertencente de cada um. Nelas possuem a opção de editá-las ou deletá-las. Além disso, possui um botão para cadastro de novas técnicas.

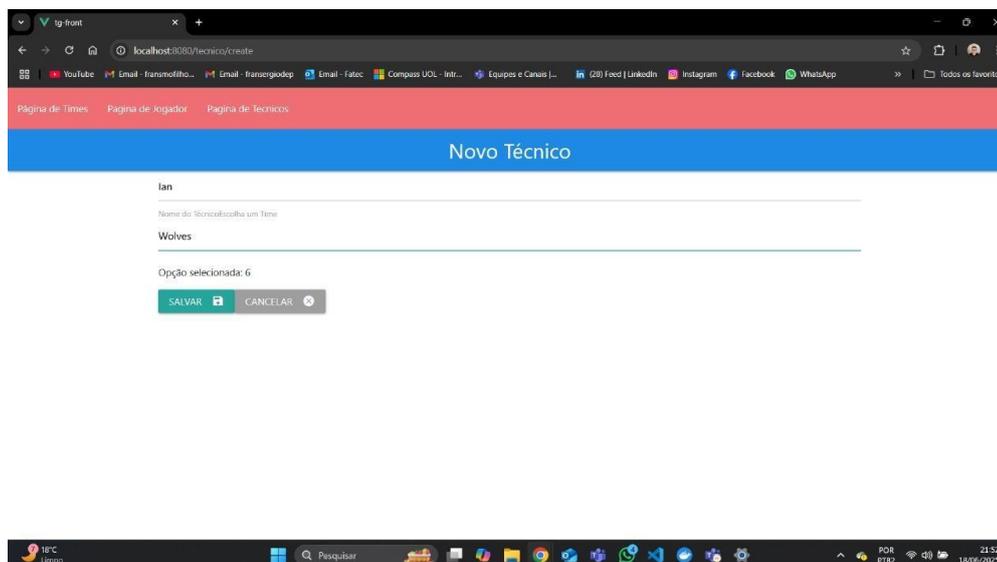
Figura 15 – Tela de Técnicos



Fonte: autoria própria

Na tela de cadastro que técnico o usuário deve preencher o nome do técnico e a equipe vinculada a ele. Após isso deve clicar em salvar para registrar.

Figura 16 – Tela de cadastro de técnicos



Fonte: autoria própria

6 Resultados e Discussão

No início do projeto, o objetivo principal era criar uma solução para controle do *scout* em jogos de vôlei, visando otimizar a coleta de dados estatísticos e melhorar a análise de desempenho de cada atleta.

Com a documentação dos requisitos funcionais e não funcionais foi possível entender quais eram as necessidades e o que deveríamos atender.

Com a criação de alguns artefatos de engenharia de software conseguimos mapear os processos e minimizar possíveis riscos da criação da solução, possibilitando identificar as funcionalidades fundamentais para o desenvolvimento do sistema.

O problema central enfrentado pelos técnicos de equipes de vôlei era a atualização e compilação dos dados da performance de cada atleta.

Este trabalho acadêmico contribui com uma solução sistêmica que possibilitará o mapeamento individual de cada atleta, através de indicadores apontados pelos técnicos e responsáveis, eliminando a necessidade de interações manuais com papéis e possibilitando que os dados sejam, obrigatoriamente, inseridos no sistema.

Isso reduz o tempo necessário para processar os gráficos e estatísticas individuais e aumenta a precisão na preparação dos treinos.

A solução se inicia na tela de login, onde diferentes tipos de acesso são disponibilizados.

Ao fazer login, o administrador do sistema poderá cadastrar as equipes, os técnicos e atletas.

Esse acesso é totalmente controlado pelo próprio usuário, oferecendo autonomia para a gestão dos atletas.

Algumas consultas através de relatórios e gráficos também foram implementadas, proporcionando aos técnicos uma base de conhecimento para tomada de decisão.

Concluimos que o objetivo principal deste trabalho acadêmico foi alcançado e desenvolvido.

6.1 Relatório Gerado via API

A seguir, é apresentado um exemplo real de relatório gerado pelo sistema de *scout* digital após uma partida de vôlei amador. Esses dados foram obtidos por meio da API integrada ao sistema e demonstram os principais recursos implementados, como registro de pontuação por jogador, estatísticas de jogo e destaque para o melhor atleta da partida.

Cofaq

13

15

Eternos

25 20 22 25 25 15

MELHOR JOGADOR

Paulo
Souza

18 PONTOS

BLOQUEIOS



9

SAQUES



4

ATAQUES



36

FONTE: API SCOUT DIGITAL

Tabela de Dados da Partida

Data da Partida: 04/07/2025 11:37

Local: Ginásio Municipal de Esportes

Partida: Cofaq x Eternos

Formato: Melhor de 5 sets (25 pontos, 5º set com 15)

Resultado Final: Eternos 3 x 2 Cofaq

Melhor Jogador: Paulo Souza – 18 pontos

Set Cofaq Eternos

| | | |
|---|----|----|
| 1 | 25 | 22 |
| 2 | 20 | 25 |
| 3 | 22 | 25 |
| 4 | 25 | 15 |
| 5 | 13 | 15 |

Scout do Melhor Jogador (Paulo Souza)

- Bloqueios: 9
- Saques: 4
- Ataques: 36

Considerações finais

Ao analisar o cotidiano dos técnicos de equipes de voleibol, onde alguns não conseguem obter parâmetros sobre a performance de cada atleta, é que surgiu a ideia de implementação de uma solução sistêmica.

Como alunos regulares do curso de Análise e Desenvolvimento de Sistemas da Fatec Franca e com a obrigatoriedade da entrega do Trabalho de Graduação, decidimos pela criação desta solução sistêmica com a devida documentação dos artefatos de Engenharia de Software.

A nossa preocupação foi preparar e implementar algo com estrutura e escopo definido e exequível, com a utilização de tecnologias pela sua eficiência, escalabilidade e suporte à comunidade, visando contribuir com este segmento esportivo tão importante e praticado no Brasil.

O sistema foi desenvolvido utilizando a biblioteca Vue.js, o que garantiu uma estrutura sólida e flexível para o desenvolvimento da interface do usuário.

A escolha dessa tecnologia proporcionou uma maior segurança no código e facilitou a criação de componentes reutilizáveis, permitindo rápidas adaptações e futuras atualizações solicitadas pelos técnicos.

Junto da integração com os serviços do SpringBoot, foi essencial para a autenticação de usuários e o armazenamento de dados, por meio um banco de dados PostgreSQL altamente escalável e adequado às demandas de sistemas.

Durante o desenvolvimento, desafios como a adaptação das interfaces para diferentes perfis de usuário (técnicos e atletas) e a implementação de funcionalidades que otimizem o processo de *scouting*.

Com o sistema em funcionamento, os técnicos podem reduzir significativamente o trabalho manual [papel] e melhorar a qualidade e a performance dos atletas.

Como funcionalidades futuras pretende-se a implementação de um painel de relatórios/gráficos gerenciais fornecendo informações detalhadas sobre a performance coletiva da equipe.

Referências

ALURA. **O que são regras de negócio?** 2021. Disponível em: <https://www.alura.com.br/artigos/o-que-sao-regras-de-negocio>.

BPMN.IO. Modelos e Negócios. 2024 Disponível em: <<https://bpmn.io>>.

CONCEITO DE. **Caso de uso** - O que é, propósito, no desenvolvimento e requisitos. 2021. Disponível em: <https://conceito.de/caso-de-uso>.

PROJECT BUILDER. **Sistema de Gestão de Projetos**. 2021. Disponível em: https://www.projectbuilder.com.br/?utm_source=google&utm_medium=cpc&utm_campaign=brand&gad_source=1&gclid=Cj0KCQjwjLGyBhCYARIsAPqTz1-yaJTr1nRhGLEB6QoIx5cUllaFNiLmkbo-AwA-Kpic3O0HDA_5lpYaAtF-EALw_wcB.

REDUX. Aplicações Java. 2024. Disponível em: <<https://redux.js.org>>.