

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Visualização de mapas de arremessos de uma partida de basquete

Leonardo Santos Abreu

PROJETO FINAL DE GRADUAÇÃO

Centro Técnico Científico – CTC
Departamento de Informática
Curso de Graduação em Engenharia da Computação

Rio de Janeiro
outubro de 2022.



Leonardo Santos Abreu

Visualização de mapas de arremessos de uma partida de basquete

Relatório de Projeto Final, apresentado ao curso de Engenharia da Computação da PUC-Rio como requisito parcial para a obtenção do título de Engenheiro de Computação.

Orientador: Waldemar Celes

Rio de Janeiro
outubro de 2022.

Agradecimentos

Gostaria de expressar minha gratidão a todas as pessoas que contribuíram direta ou indiretamente para a conclusão deste trabalho. Sem o apoio delas, este projeto teria sido ainda mais difícil.

Agradeço ao meu orientador, Waldemar Celes, por sua orientação valiosa, apoio contínuo e paciência durante todo o processo de pesquisa e redação deste trabalho. Suas sugestões e conselhos foram fundamentais para o seu sucesso.

Agradeço aos meus amigos, que compartilharam suas ideias, conhecimentos e experiências comigo ao longo desta jornada acadêmica. Agradeço pela troca de informações, pelas discussões enriquecedoras e pelo apoio mútuo em momentos de dificuldade.

Agradeço aos meus pais e familiares, cujo amor, incentivo e apoio incondicional foram essenciais para a minha educação e desenvolvimento. Sem eles, eu não estaria aqui hoje.

Agradeço a todos os professores, bibliotecários e funcionários da instituição de ensino Pontifícia Universidade Católica do Rio de Janeiro, que proporcionaram um ambiente acadêmico propício para a realização deste estudo.

Meus sinceros agradecimentos a todos que, de alguma forma, contribuíram para este trabalho e para a minha formação acadêmica.

Leonardo Santos Abreu

Resumo

Abreu, Leonardo. Celes, Waldemar. Visualização de mapas de arremessos de uma partida de basquete. Rio de Janeiro, ano 2023. 43 p. Relatório de Projeto Final – Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

Este projeto se concentra na investigação e implementação de soluções para os desafios associados à visualização de um grande volume de dados em um mapa específico. Para atingir esse objetivo, optamos por aplicar nosso estudo a eventos de uma partida de basquete, considerando a complexidade existente na compreensão visual de todos os arremessos realizados. O projeto busca encontrar métodos eficientes de visualização por meio de estudos e práticas utilizadas em outras áreas da visualização. Como conclusão, ao longo deste trabalho foram implementados métodos de visualização que buscam facilitar a observação e compreensão de todos os arremessos de uma ou mais partidas de basquete, facilitando a análise e tomada de decisões de profissionais da área.

Palavras-chave: Visualização de dados; Basquete; Mapas de calor; Mapas de calor com áreas customizadas; Mapas de pontos; Mapas de pontos sem sobreposição; Ruído de dados

Abstract

Abreu, Leonardo. Celes, Waldemar. Basketball shots map visualization. Rio de Janeiro, 2023. 43 p. Final Project Report – Department of Informatics. Pontifícia Universidade Católica do Rio de Janeiro.

This project focuses on the research and implementation of solutions for the challenges associated with visualizing a large volume of data on a specific map. To achieve this goal, we've chosen to apply our study to events from a basketball game, considering the complexity in the visual understanding of all the shots taken. The project main goal is to find efficient visualization methods through studies and practices used in other areas of visualization. In conclusion, throughout this work, visualization methods were implemented that aim to facilitate the observation and understanding of all shots from one or more basketball games, making the analysis and decision-making process easier for professionals in the field.

Keywords: Data Visualization; Basketball; Heatmaps; Customized heatmaps; Scatter plot; Scatter plot without overlapping; Data noise

Sumário

Agradecimentos	3
Resumo.....	4
Abstract.....	5
Lista de figuras.....	7
Introdução	10
Ambiente Computacional.....	11
Situação atual	11
Objetivos	13
Atividades realizadas	13
Estudos preliminares	13
Estudos conceituais e de tecnologia.....	13
Desenvolvimento	16
Testes e Protótipos para aprendizado e demonstração	17
Metodologia	17
Projeto e especificação do sistema	18
Obtenção dos dados.....	18
Exibição dos eventos.....	19
Visualização de círculos.....	21
Mapa de calor	22
Mapa de calor com polígonos customizados.....	25
Mapa de círculos agrupados	31
Visão geral.....	35
Sistema	35
Testes.....	35
Implementação e avaliação.....	36
Planejamento e execução de testes funcionais	36
Planejamento e execução de outros testes.....	42
Comentários sobre a implementação.....	42

Considerações finais	42
Contribuições.....	42
Aprendizado	43
Projetos Futuros	44
Referências Bibliográficas.....	45
Apêndices	47
Manual de usuário	47
Introdução.....	47
Visão geral.....	47
Requisitos do sistema.....	47
Instalação e execução.....	47
Primeiros passos.....	47

Lista de figuras

Figura 1: Exemplo de como era feita a visualização customizada de eventos antes do projeto	12
Figura 2: Exemplo de como era feita a visualização de eventos antes do projeto ...	12
Figura 3: Comunicação com a API	14
Figura 4: Fluxo de requisições para a API durante o preenchimento do formulário .	15
Figura 5: Diagrama do fluxo da aplicação	16
Figura 6: Teste da comunicação com a API	17
Figura 7: Formulário	19
Figura 8: Dados sem espelhamento.....	20
Figura 9: Dados com espelhamento.....	20
Figura 10: Mapa de círculos	21
Figura 11: Mapa de círculos de todas as partidas da temporada de 2021	22
Figura 12: Escala utilizada nos mapas de calor.....	23
Figura 13: Exemplo de como o usuário visualiza a escala ao lado da quadra	23

Figura 14: Mapa de calor.....	24
Figura 15: Mapa de calor de todas as partidas da temporada de 2021	24
Figura 16: Caixas de informações do mapa de calor de todas as partidas da temporada de 2021	25
Figura 17: Visualização de zonas utilizada pela NBA	26
Figura 18: Polígonos desenhados no Illustrator.....	27
Figura 19: Polígonos desenhados no Illustrator aplicados na quadra	27
Figura 20: Escala utilizada nos mapas de calor.....	28
Figura 21: Exemplo de como o usuário visualiza a escala ao lado da quadra	28
Figura 22: Exemplo de mapa de calor customizado	30
Figura 23: Mapa de calor customizado de todas as partidas da temporada de 2021	30
Figura 24: Caixa de informações do mapa de calor customizado de todas as partidas da temporada de 2021.....	30
Figura 25: Mapa de círculos agrupados	34
Figura 26: Mapa de círculos agrupados para todas as partidas da temporada de 2021	34
Figura 27: Comparação entre a exibição sem agrupamento (à esquerda) e com agrupamento (à direita) para as partidas FRA x FLA (09/06/2022), FLA x FRA (04/06/2022), FLA x FRA (02/06/2022), FRA x FLA (28/05/2022) e MIN x FLA (17/05/2022).....	34
Figura 28: Requisição realizada pelo envio do formulário	37
Figura 29: Requisição feita pelo postman.....	38
Figura 30: Arremesso de dois pontos errado no limite do garrafão indicado pelo mouse	39
Figura 31: Arremesso de três pontos errado logo após o limite do garrafão indicado pelo mouse	39
Figura 32: Visualização de círculos dos arremessos dos jogadores Rafael, Carlos, Brandon e Tulio na partida de 09/06/2022.....	40
Figura 33: Comparação da Visualização de círculos (à esquerda) com a Visualização agrupada (à direita) dos arremessos dos jogadores Rafael, Carlos, Brandon e Tulio na partida de 09/06/2022.....	40

Figura 34: Comparação da Visualização de círculos (à esquerda) com o Mapa de calor (à direita) dos arremessos dos jogadores Rafael, Carlos, Brandon e Tulio na partida de 09/06/2022	40
Figura 35: Comparação da Visualização de círculos (à esquerda) com o Mapa de calor customizado (à direita) dos arremessos dos jogadores Rafael, Carlos, Brandon e Tulio na partida de 09/06/2022	41
Figura 36: Comparação entre o critério de menor raio (à esquerda) e da maior sobreposição (à direita) das partidas FRA x FLA (09/06/2022), FLA x FRA (04/06/2022), FLA x FRA (02/06/2022), FRA x FLA (28/05/2022) e MIN x FLA (17/05/2022).....	41

Introdução

A visualização de dados é a apresentação de informações em formato gráfico e permite que tomadores de decisões vejam os resultados de suas análises visualmente. De acordo com o CEO da Infogram, Mikko Jarvenpaa[1], pessoas melhores informadas tomam decisões melhores, e as pessoas que podem ler e criar comunicações baseadas em dados são fundamentais para isso.

Apesar da visualização de dados ter um papel essencial na tomada de decisões, existem muitos cuidados a serem tomados ao tornar todos os seus dados visíveis, especialmente quando o volume de dados é muito grande. Esse cenário fica ainda mais complicado quando necessitamos reduzir e posicionar todo esse volume de informação em um mapa e, quanto menor o tamanho desse mapa, mais complicações podem surgir.

Imagine visualizar um plano cartesiano onde cerca de 60% dos pontos se encontram em uma área correspondente a apenas 35% do tamanho total deste plano, já é de difícil visualização. Agora, imagine o mesmo cenário onde a cor de cada ponto, por exemplo, seja de extrema relevância para a interpretação deste plano e, diante desse cenário, já chegamos em um gráfico possivelmente incompreensível. Essa dificuldade em interpretar os dados é um problema comum conhecido como "ruído de dados", conforme abordado no jornal da UFRP [2]. O ruído de dados representa conteúdos presentes nas bases de dados que podem prejudicar a qualidade das informações extraídas, resultando em perda de informação. Esses são problemas comuns na visualização de grandes quantidades de dados.

Ao fazer uma pesquisa, podemos ver diversas soluções para os problemas listados acima, já que são problemas enfrentados diariamente na exibição de grande quantidade de dados, de acordo com o artigo publicado em SCIEPUB [3]. Pensando nisso, surgiu a ideia de aplicar todo esse conhecimento na visualização de eventos de uma partida de basquete.

Durante uma partida de basquete, são registrados cerca de 200 eventos para serem exibidos em um mapa da quadra. Para que todos esses eventos agreguem valor na análise dos profissionais, são registrados alguns dados junto com a posição e tipo de evento. Os arremessos de dois pontos, por exemplo, carregam com o evento o Jogador, Time, Posição e Placar, e todas essas informações são relevantes e devem ser exibidas de alguma forma na visualização. Torna-se evidente que a

visualização de todos esses dados necessita de um tratamento específico. O objetivo desse trabalho é tornar todos esses dados compreensíveis e intuitivos para facilitar e melhorar a experiência dos profissionais que utilizam essas informações para tomar decisões em cima do elenco de um time.

Ambiente Computacional

Para iniciar o desenvolvimento do Projeto, foi utilizado um computador pessoal com Sistema Operacional Windows. Além disso, o site está sendo desenvolvido em Typescript, por meio do Framework Angular e suas bibliotecas.

Situação atual

A visualização de todos os eventos de uma partida de basquete específica foi desenvolvida em MATLAB pelo instituto Tecgraf. Atualmente, é possível visualizar todos os dados agrupados no mapa de uma quadra por meio de círculos desenhados na localização dos eventos. Todos esses dados são exibidos sem muitos recursos para auxiliar a visualização e o entendimento do usuário. A ideia é tornar esses dados legíveis em uma plataforma Web desenvolvida em Typescript, por meio do framework Angular.

Os dados são exibidos de forma simples e, por esse motivo, a interpretação dos dados não é de fácil compreensão e, além disso, pode gerar interpretações erradas em alguns contextos. Essa situação é comum e, por isso, existem diversos algoritmos e estudos nessa área que buscam facilitar o entendimento de grande quantidade de dados em cenários específicos. As figuras 1 e 2 buscam ilustrar como a visualização pode parecer confusa olhando para a visualização atual dos eventos.

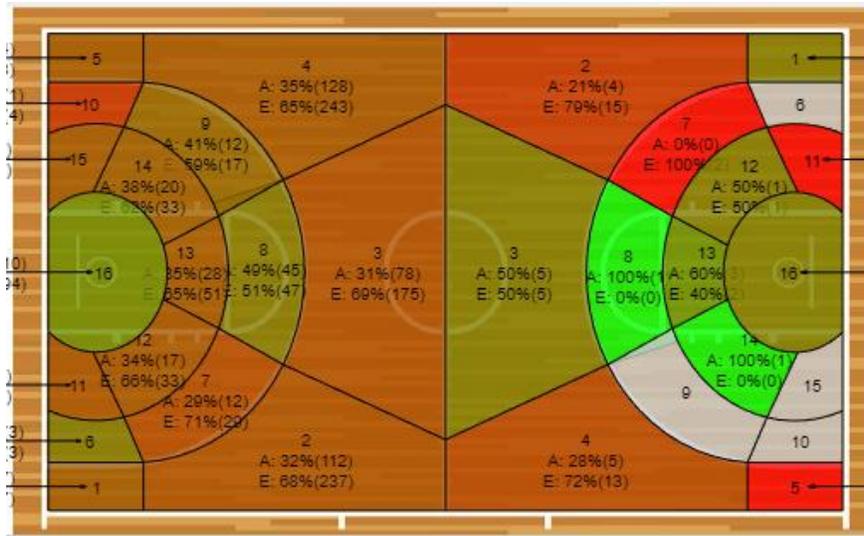


Figura 1: Exemplo de como era feita a visualização customizada de eventos antes do projeto

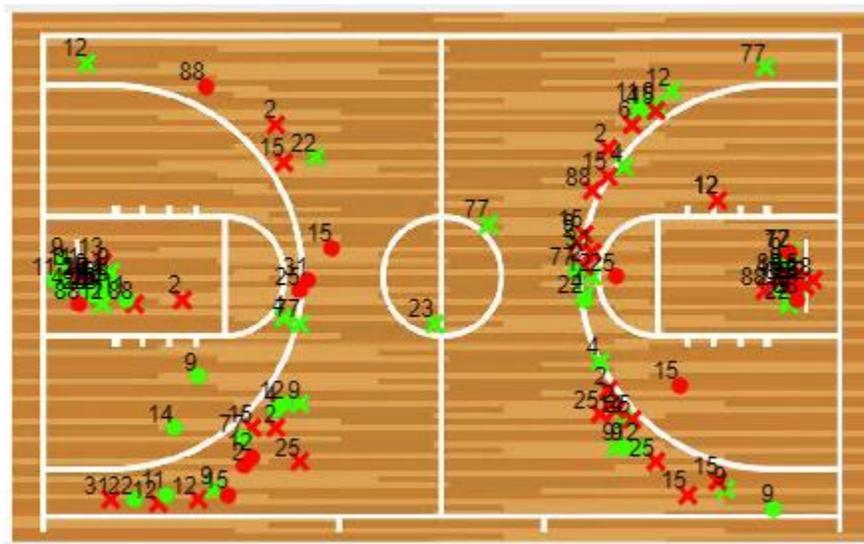


Figura 2: Exemplo de como era feita a visualização de eventos antes do projeto

Como todas essas informações são de extrema importância para a tomada de decisões dentro da equipe, precisamos buscar uma maneira de exibir isso da melhor forma possível. Esse projeto busca suprir essas necessidades, por meio da implementação de algoritmos de visualização, facilitando o entendimento e a tomada de decisões dos profissionais da área.

O projeto em questão, busca solucionar essa questão citada anteriormente. Para isso, foram desenvolvidas quatro visualizações diferentes, cada uma apresentando pontos fortes e fracos distintos.

Objetivos

A ideia é apresentar os dados oferecidos pela API do Tecgraf em uma plataforma Web, para que assim, facilite o acesso e a compreensão. Para isso, foi escolhido o Framework Angular[4], que é utilizado por muitas empresas no mercado como em sites Google Cloud [5], McDonalds [6], Apple Support [7], Forbes [8]. Além disso, esse framework ajuda a construir aplicações de página única de maneira interativa e dinâmica, justamente o que buscamos fazer para a visualização de dados. O projeto será desenvolvido em Typescript, que é a linguagem na qual Angular foi baseado.

Hoje todos os dados são posicionados em uma representação da quadra, mas sem nenhuma forma de tratamento no MATLAB. Como foi desenvolvido em um software voltado para cálculos numéricos, é difícil de fazer manutenção ou implementar melhorias na visualização desses dados. O objetivo do trabalho é criar uma plataforma com diversas alternativas de visualizações e organizar esses dados, além de possibilitar e facilitar futuras implementações e melhorias.

A solução busca se diferenciar pela maneira na qual os dados serão exibidos para o usuário, seja por meio de diversos algoritmos de visualização ou por maneiras de separar os dados fornecidos, como visualizar todos os arremessos convertidos de um jogador específico, por exemplo. Todas essas opções seriam de extrema importância para todos os profissionais que fazem uso dessas informações conseguirem tomar decisões mais precisas e de forma mais fácil.

Atividades realizadas

Estudos preliminares

Para o início do desenvolvimento do projeto, foi aproveitado o conhecimento prévio sobre desenvolvimento web para a criação e ideação inicial do servidor. Além disso, a ambientação com o uso de APIs facilitou o entendimento e o desenvolvimento da chamada de rotas por meio de um formulário.

Estudos conceituais e de tecnologia

Inicialmente foi necessário estudar a melhor forma de realizar a comunicação entre o servidor web e a API do Tecgraf, responsável pela consulta dos eventos de basquete. Para isso, foi pensada uma comunicação inicial bem simples como forma de teste, assim como é ilustrado na figura 1.

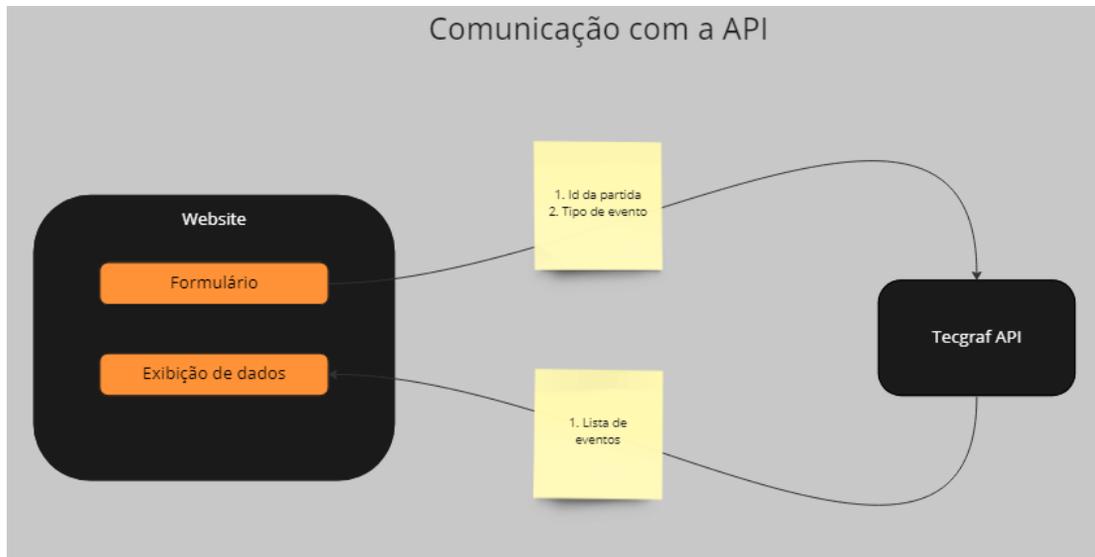


Figura 3: Comunicação com a API

Além disso, foi necessário realizar um estudo sobre a como estruturar uma aplicação Angular. Isso porque apesar do conhecimento de desenvolvimento web ajudar, esse framework tem algumas particularidades que podem tornar o início do aprendizado um pouco confuso. Logo, para evitar complicações futuras geradas pela falta de conhecimento prévio, antes do desenvolvimento foi realizado um estudo sobre como organizar a aplicação de forma inteligente.

Para complementar o estudo anterior, foi iniciado o desenvolvimento de um formulário simples em Angular para iniciar a ambientação com o Framework e testar o conhecimento adquirido anteriormente. Esse passo foi complementado utilizando os dados desse formulário para realizar a chamada da API, assim como o diagrama acima mostra. Depois de estudar sobre Typescript, foi criada uma interface `ApiEvent`, responsável por garantir o acesso prático aos eventos que a API entrega ao servidor.

Ao iniciar o desenvolvimento do projeto, foi pensado um fluxo estruturado para otimizar o preenchimento do formulário encarregado de solicitar os eventos à API. Este fluxo é ilustrado na figura 2, onde cada bloco à esquerda simboliza o estímulo para a requisição representada pelas setas de cor vermelha.

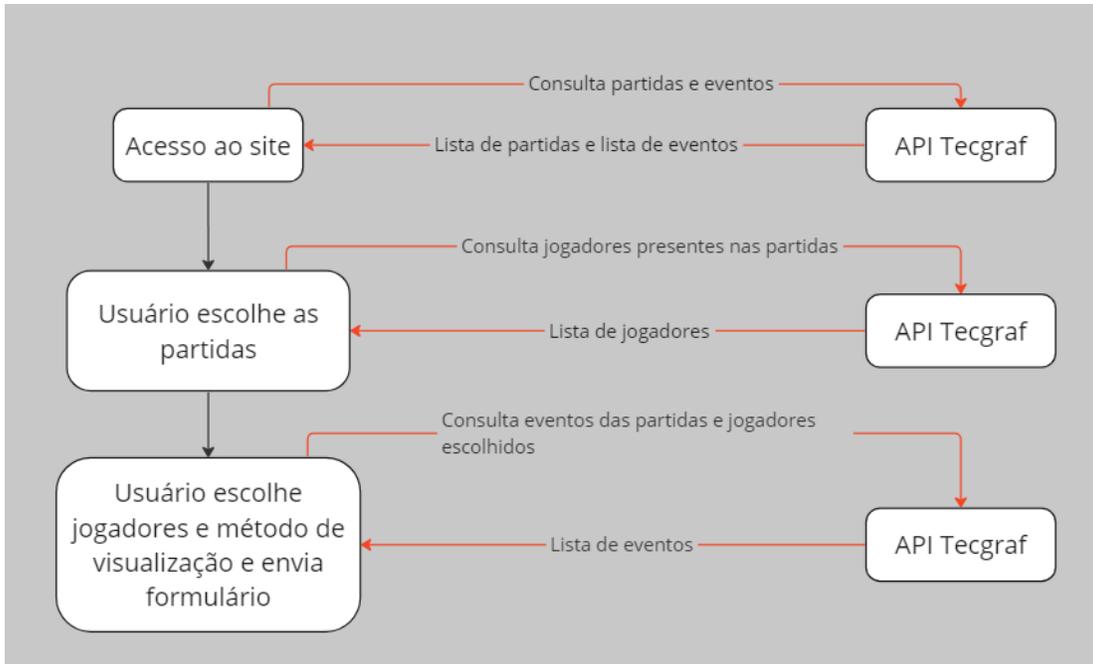


Figura 4: Fluxo de requisições para a API durante o preenchimento do formulário

Além disso, efetuou-se um estudo para delimitação da arquitetura da aplicação, possibilitando que o desenvolvimento se iniciasse com a estrutura já preparada, a fim de prevenir retrabalho futuro. Com essa finalidade, elaborou-se um diagrama destinado a representar o fluxo de percurso dos dados da aplicação até a sua exibição. Este diagrama está ilustrado pela figura 3 e o número anterior ao texto indica a ordem do acontecimento, percorrendo do 1 até o 7.

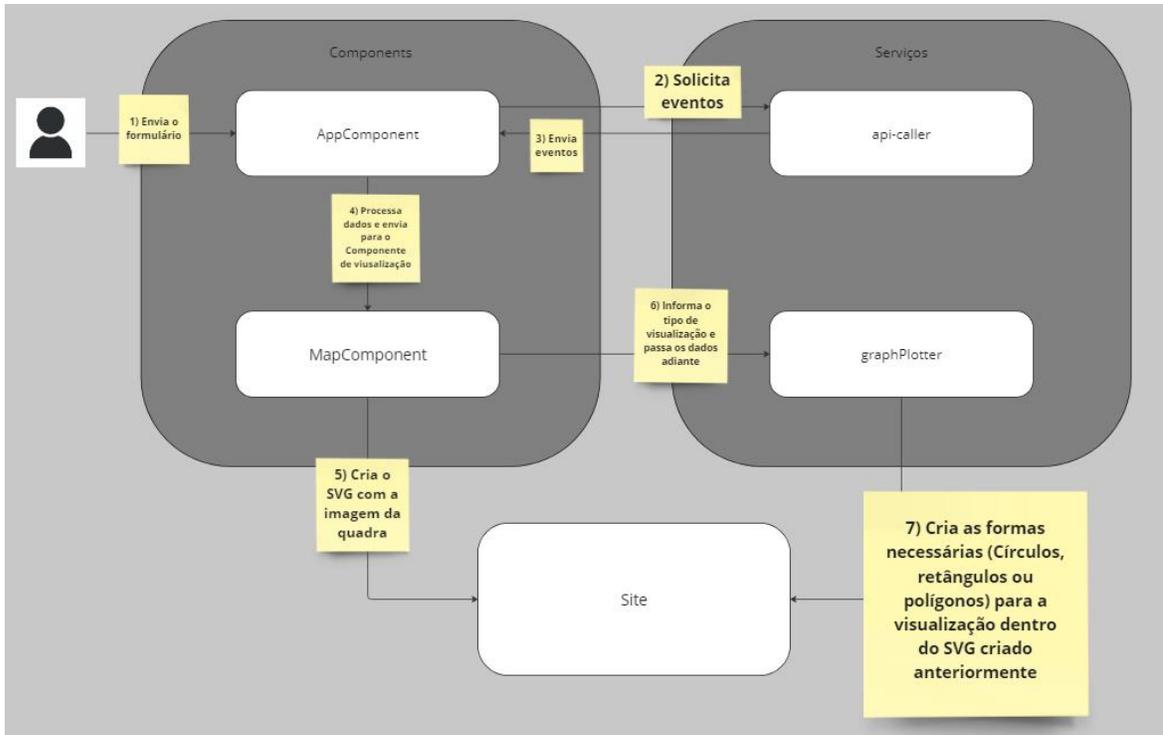


Figura 5: Diagrama do fluxo da aplicação

Após a concretização de um esboço integral da aplicação, deu-se início no desenvolvimento da aplicação, começando pela implementação do formulário. Este elemento é crucial, uma vez que representa a interface principal para a interação do usuário com a aplicação, permitindo a entrada de dados que orientarão as operações subsequentes.

Desenvolvimento

Com a construção do formulário, a fase seguinte do desenvolvimento consistiu na criação das quatro distintas opções de visualização. Esta etapa exigiu um trabalho meticuloso, dada a necessidade de garantir que cada visualização fosse não apenas funcional, mas também intuitiva e informativa.

Assim, foram realizadas requisições para as mesmas partidas, buscando extrair informações relevantes para a equipe técnica. Observamos que, em cenários específicos, algumas visualizações se destacaram. No entanto, no geral, a visualização de pontos agrupados apresentou o melhor resultado.

Além das pesquisas mencionadas, foram necessárias investigações adicionais para compreender a metodologia mais eficiente de processamento de dados e sua representação adequada em diferentes visualizações. Encontramos

desafios consideráveis na apresentação de polígonos personalizados, exigindo uma exploração aprofundada sobre diferentes maneiras de exibí-los no mapa. Depois de um tempo considerável de pesquisa, concluiu-se que a melhor forma de realizar essa exibição seria por meio de um SVG (Scalable Vector Graphics), um formato de arquivo vetorial compatível com a internet, conforme descrito pela Adobe [9], e a exibição de formas poligonais.

Ademais, a visualização de agrupamentos circulares proporcionou aprendizado significativo, dado que foi necessário realizar uma pausa para buscar alternativas para o agrupamento de tais dados. Após uma avaliação criteriosa, concluímos que a implementação do Algoritmo Guloso se configuraria como a opção mais apropriada para o cenário em questão.

Testes e Protótipos para aprendizado e demonstração

Para iniciar os testes e ambientação da aplicação, foi desenvolvido um formulário capaz de chamar uma rota específica da API depois de preenchido. Os dados dos campos estão sendo utilizados para testes por enquanto, mas futuramente o usuário não irá precisar saber o ID da partida e nem o código do evento. Na figura 4, é possível visualizar a resposta da API de 76 eventos ao buscar pela partida 23977 e evento A2C.

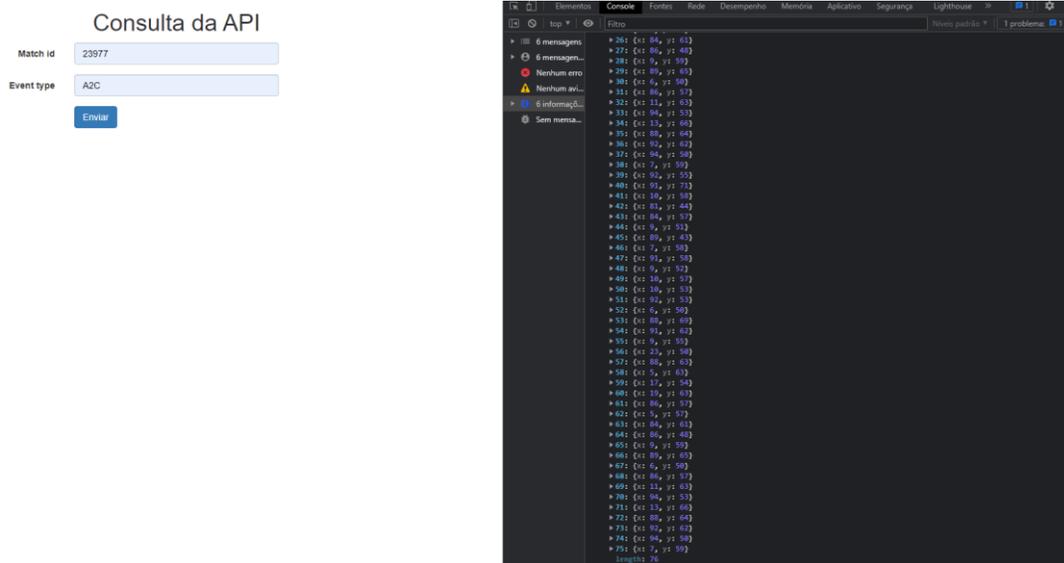


Figura 6: Teste da comunicação com a API

Metodologia

No curso do desenvolvimento do projeto, estabeleceu-se um ritmo de reuniões quinzenais com o orientador. Estas sessões serviam como um fórum para

a identificação e distribuição de tarefas a serem executadas durante a fase de desenvolvimento subsequente.

A abordagem de gerenciamento de projeto adotada durante esse processo assemelha-se notavelmente à metodologia Kanban, um sistema de gerenciamento visual de projetos que busca otimizar fluxos de trabalho. De forma rotineira, as tarefas eram estratificadas e alocadas, com a progressão do desenvolvimento seguindo uma ordem de prioridade estabelecida com base nos tópicos levantados em cada reunião.

Projeto e especificação do sistema

O sistema desenvolvido consiste em uma aplicação web que utiliza Typescript e o framework Angular. O projeto é estruturado em dois componentes, um deles é responsável pelo gerenciamento do formulário e o outro pelo mapa e a apresentação dos eventos. Além desses componentes, a aplicação possui serviços que atuam como intermediários, fornecendo funcionalidades cruciais para o bom funcionamento do sistema. Um desses serviços é o api-caller, que tem a responsabilidade de fazer as chamadas para a API do Instituto Tecgraf. Este serviço se encarrega de todas as comunicações entre a aplicação e a API, assegurando que os dados sejam transmitidos corretamente e as respostas sejam tratadas de forma adequada.

Obtenção dos dados

Para obter os dados necessários, a aplicação utiliza variáveis de ambiente contendo o Token e a URI, que permitem o acesso à API. Inicialmente a aplicação consulta a API para conseguir a lista de partidas e os eventos disponíveis para consulta. Depois disso, é feita a requisição para conseguir a lista de todos os jogadores presentes nas partidas escolhidas pelo usuário. Por fim, o sistema realiza uma última consulta para conseguir a lista dos eventos realizados pelos jogadores dentro das partidas que ele escolheu. Na figura 5, é possível visualizar uma imagem do formulário preenchido por um usuário.

Consulta da API

Partida

Tipo de evento

Jogadores

Visualização

Quantidade de eventos

Figura 7: Formulário

Exibição dos eventos

Para exibir os dados resgatados na API, a aplicação precisa realizar um tratamento especial para facilitar o entendimento da origem dos eventos pelos usuários. Isso ocorre porque a troca de lados por quarto no basquete é uma prática comum e serve para garantir a equidade e a imparcialidade na competição, considerando que fatores como a posição dos bancos de reservas e a orientação das equipes em relação à quadra podem influenciar no desempenho.

Ao longo da pesquisa e do desenvolvimento da aplicação, concluímos que o espelhamento dos dados seria a forma mais intuitiva de lidar com essas trocas. Como o projeto é em parceria com o Flamengo, entendemos que uma boa divisão seria se todos os eventos originados pelos jogadores do Flamengo ficassem no lado esquerdo do mapa, enquanto os de seus adversários ficassem no direito.

Por isso, uma vez que o usuário envia o formulário preenchido, a aplicação realiza o espelhamento de dados, garantindo a divisão de eventos citada anteriormente. Para ilustrar o funcionamento desse tratamento, foram separadas duas imagens da visualização dos arremessos do Flamengo no jogo FRA x FLA (2022-06-09). A figura 6 representa a visualização antes do espelhamento, já a 7 mostra os dados já tratados. Em ambas as imagens a seguir, o lado esquerdo

representa os arremessos realizados pelo Flamengo, enquanto o lado direito mostra os arremessos dos adversários.

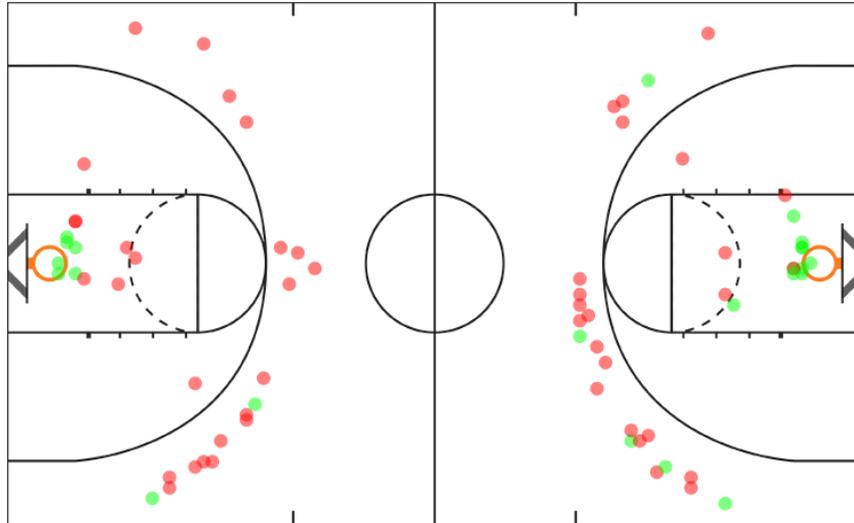


Figura 8: Dados sem espelhamento

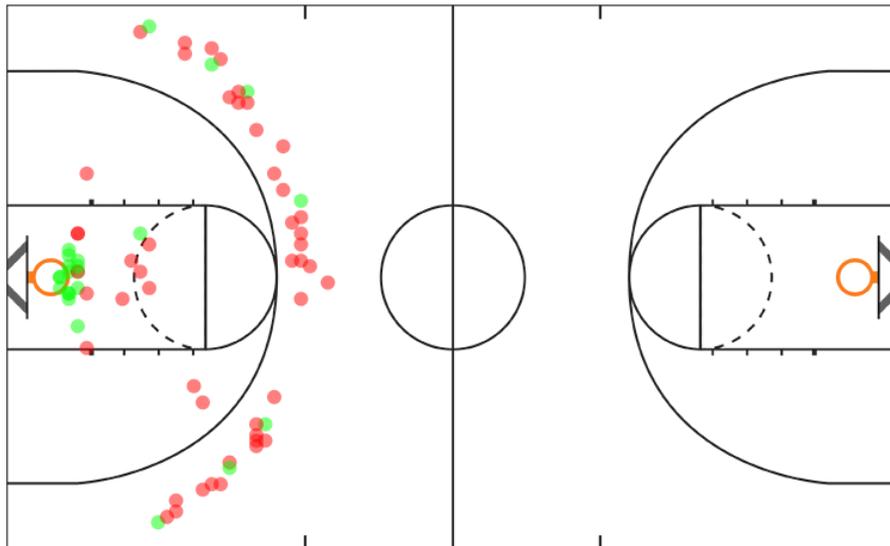


Figura 9: Dados com espelhamento

Depois do espelhamento, os dados são organizados conforme o tipo de visualização escolhida pelo usuário. Cada visualização tem um tratamento necessário para que os dados sejam exibidos da forma estudada e proposta, posteriormente iremos entender um pouco melhor de como cada visualização precisa ser tratada.

Visualização de círculos

Nesse caso em específico, o componente de exibição só precisa das coordenadas de cada evento para exibir o mapa completo com os eventos, por esse motivo, não tem nenhum tratamento especial realizado depois do espelhamento para esse tipo de mapa.

Após receber os dados, o componente então cria uma visualização de pontos onde a posição, o tamanho e a cor dos pontos são determinados pelos dados. Além disso, ele adiciona interatividade para que, quando o usuário colocar o ponteiro sobre um círculo, ele exiba uma caixa com mais informações. A figura 8 busca representar essa maneira de visualizar os eventos. Na ilustração abaixo, o lado esquerdo representa os arremessos realizados pelo Flamengo, enquanto o lado direito mostra os arremessos dos adversários.

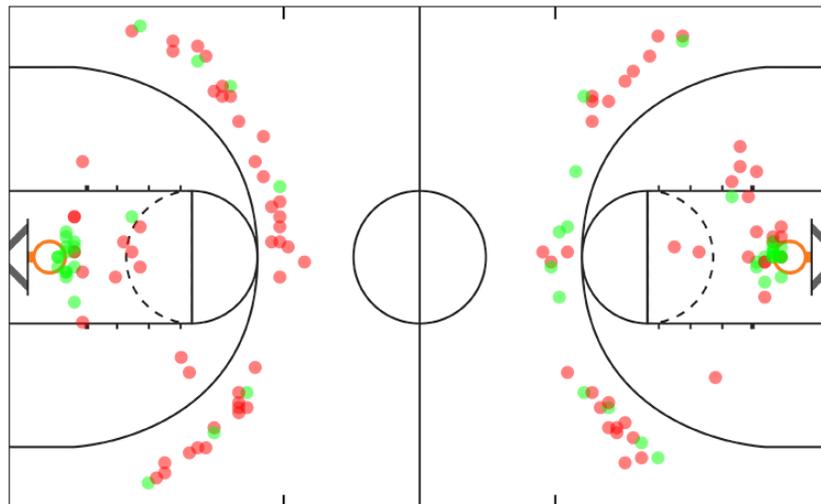


Figura 10: Mapa de círculos

A ilustração acima evidencia que esta visualização se destaca pela alta precisão na localização dos eventos e pela utilização cores para indicar acertos e erros, mostrando uma visualização muito intuitiva para uma quantidade pequena de eventos. No entanto, essa abordagem possui alguns pontos negativos em cenários um pouco mais reais, como a grande quantidade de dados agrupados (ruído visual), a dificuldade de compreensão da taxa de conversão geral dos atletas e da quantidade de arremessos em determinados locais do mapa, além do fato de que eventos diferentes sobrepostos podem causar confusão aos usuários. A figura 9 busca ilustrar a dificuldade encontrada para visualizar todos os arremessos de todas as partidas da temporada de 2021.

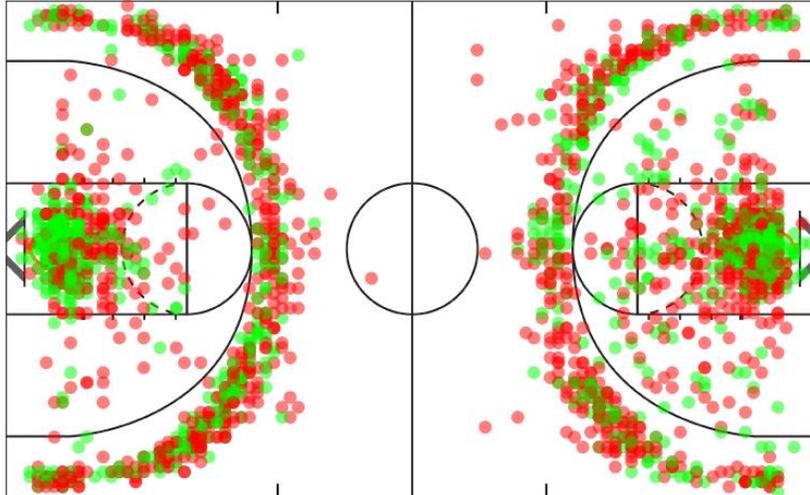


Figura 11: Mapa de círculos de todas as partidas da temporada de 2021

Ao analisar a imagem acima, torna-se evidente a complexidade de interpretação, o que motivou o desenvolvimento de outros métodos de visualização para outros contextos. A estratégia de espelhamento de dados superou um desafio significativo, no entanto, a visualização ainda enfrenta obstáculos ao lidar com um volume elevado de dados devido à presença de ruído.

Mapa de calor

Em consequência do problema observado com os testes da Visualização de círculos tradicional, foram conduzidas investigações com o objetivo de identificar técnicas adicionais que possam tratar eficientemente essa volumetria de maneira intuitiva. Durante essa pesquisa, a conclusão obtida é que os mapas de calor seriam uma alternativa viável para abordar esse desafio. Tal escolha é fundamentada na perspectiva apresentada na publicação "Nature Maps", onde é afirmado que "A representação densa e intuitiva torna os mapas de calor apropriados para a apresentação de dados de alto volume" (Nature Maps, 2023) [10].

Após avaliarmos diferentes aplicações e implementações de mapas de calor, optamos por uma versão mais avançada desse recurso. Criamos um mapa de calor que exibiu simultaneamente a taxa de conversão de arremessos (indicada pela cor) e a quantidade de arremessos (representada pela opacidade).

Após a realização de múltiplos testes com diversas escalas, chegou-se à conclusão de que seria apropriado adotar uma escala que transitasse do Vermelho, representando uma taxa de conversão baixa, até o Verde, simbolizando uma taxa de conversão alta. Por essa razão, optou-se por uma escala que preenche exatamente estes critérios, e ela pode ser visualizada na figura 10.



Figura 12: Escala utilizada nos mapas de calor

Sempre que o usuário opta por essa visualização, essa escala é ao lado do mapa da quadra de basquete para ajudar o usuário a entender melhor as cores dos retângulos. A figura 11 ilustra como o usuário observa essa informação.

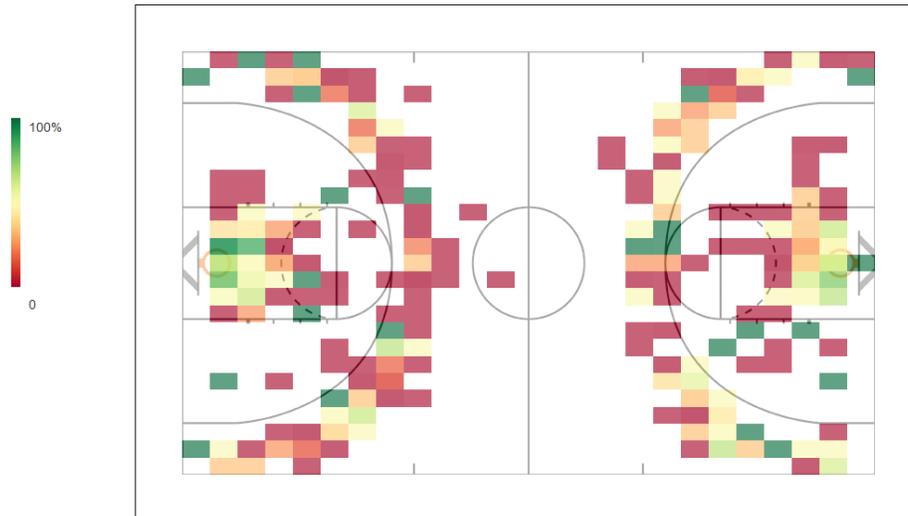


Figura 13: Exemplo de como o usuário visualiza a escala ao lado da quadra

Nesta visualização, cada arremesso é atribuído a uma célula específica dentro de uma matriz bidimensional, que serve como nosso mapa de calor. O mapeamento da posição do arremesso para uma célula específica na matriz é determinado por uma escala, cujo tamanho é ditado pelo valor da variável *heatmapSize*. Cada célula desta matriz armazena informações sobre a quantidade de arremessos realizados dentro da área que ela representa no mapa. Além disso, cada uma delas também armazena o número de arremessos que resultaram em acertos e erros, permitindo que esses dados sejam usados mais para frente na visualização.

Essa matriz é enviada para o componente de exibição, que é responsável por desenhá-la, montando um mapa de calor de arremessos em um jogo. Para isso, a aplicação vai desenhar um retângulo para cada elemento da matriz recebida, mas cada retângulo terá características diferentes. A cor de cada retângulo é determinada pela taxa de conversão de arremessos (acertos / total de arremessos), que é mapeada para uma escala de cores do vermelho ao verde, passando pelo amarelo, onde 0% é vermelho e a maior taxa de conversão de todos os retângulos é o limite superior da escala. Além disso, a opacidade de cada retângulo varia de 0,6 a 1, isso é feito por uma escala determinada pelo número de arremessos de cada célula no

mapa de calor. O domínio da escala é de 0 até o maior número de arremessos em qualquer célula. No entanto, um tratamento especial é aplicado às células sem arremessos, ajustando a opacidade para 0, tornando-as completamente transparentes.

As figuras a seguir objetivam exemplificar a visualização discutida acima. A figura 12 representa o cenário de um único jogo, enquanto a 13 engloba todos os arremessos de todos os jogos disponíveis na API. Em todas as imagens a seguir, o lado esquerdo representa os arremessos realizados pelo Flamengo, enquanto o lado direito mostra os arremessos dos adversários.

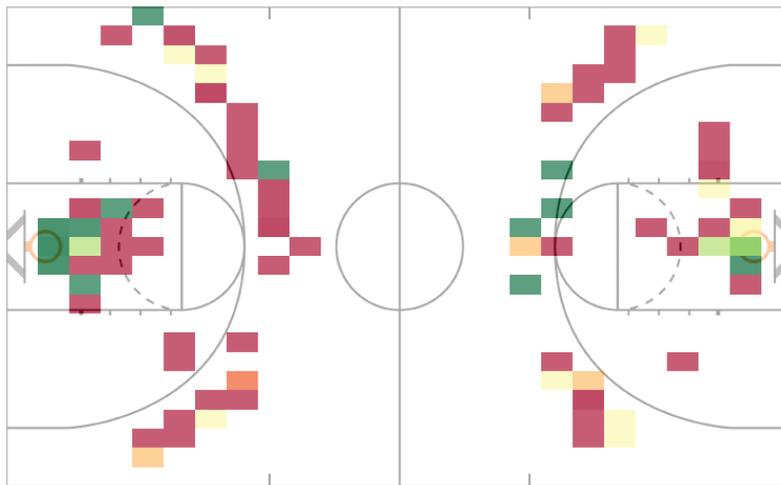


Figura 14: Mapa de calor

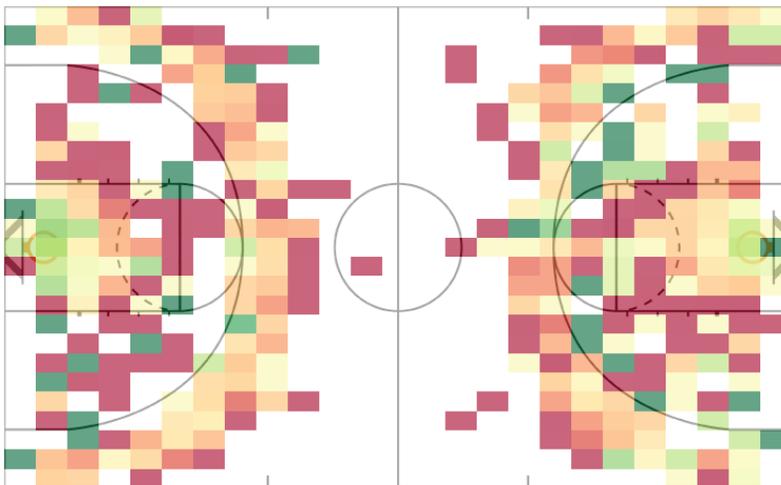


Figura 15: Mapa de calor de todas as partidas da temporada de 2021

A figura acima evidencia que essa visualização oferece uma grande facilidade na observação e no entendimento, já que facilita a identificação de áreas com alta ou baixa conversão. Porém, é importante mencionar que essa visualização não possui

uma escala tão intuitiva e nem a mesma precisão que o mapa de pontos. Além disso, para representar alguns dados, como taxa de conversão, ocorrências e quantidade de eventos, é necessário recorrer a caixas de informações. A figura 14 busca ilustrar algumas informações que as caixas de informações trazem para facilitar o entendimento do mapa.

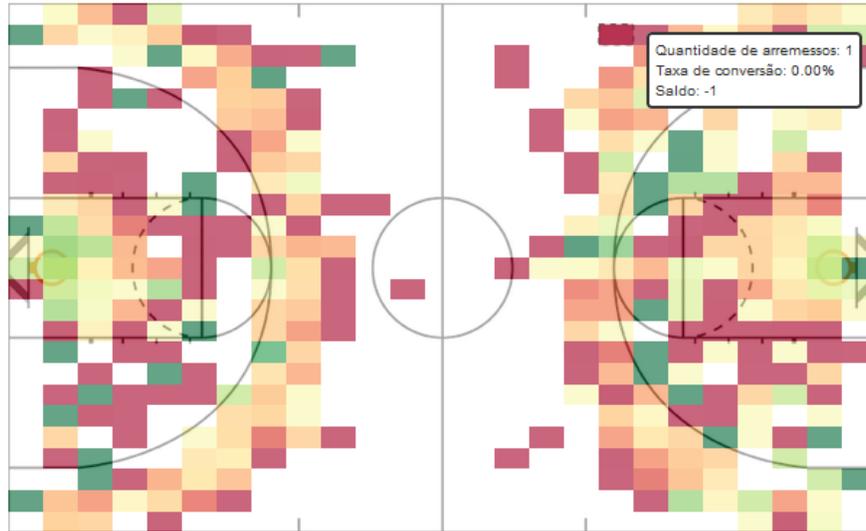


Figura 16: Caixas de informações do mapa de calor de todas as partidas da temporada de 2021

Para além das dificuldades mencionadas anteriormente, fica claro que os retângulos, em sua forma atual, não fornecem informações tão significativas quando se analisa a quadra.

Mapa de calor com polígonos customizados

A falta de clareza no Mapa de calor tradicional instigou uma pesquisa adicional, a partir da qual reconhecemos a possibilidade de implementar uma visualização por meio de polígonos personalizados - uma técnica já empregada na análise de arremessos de basquete. Essas áreas são comumente utilizadas em visualizações de arremessos porque separam as próprias marcações da quadra de basquete em zonas mais intuitivas, permitindo que os jogadores, técnicos ou analistas consigam analisar e identificar rapidamente a efetividade e as tendências de arremessos. A NBA, por exemplo, emprega essa metodologia em sua plataforma de visualização de arremessos durante uma partida, disponível no site oficial [11], como é possível observar na figura 15.

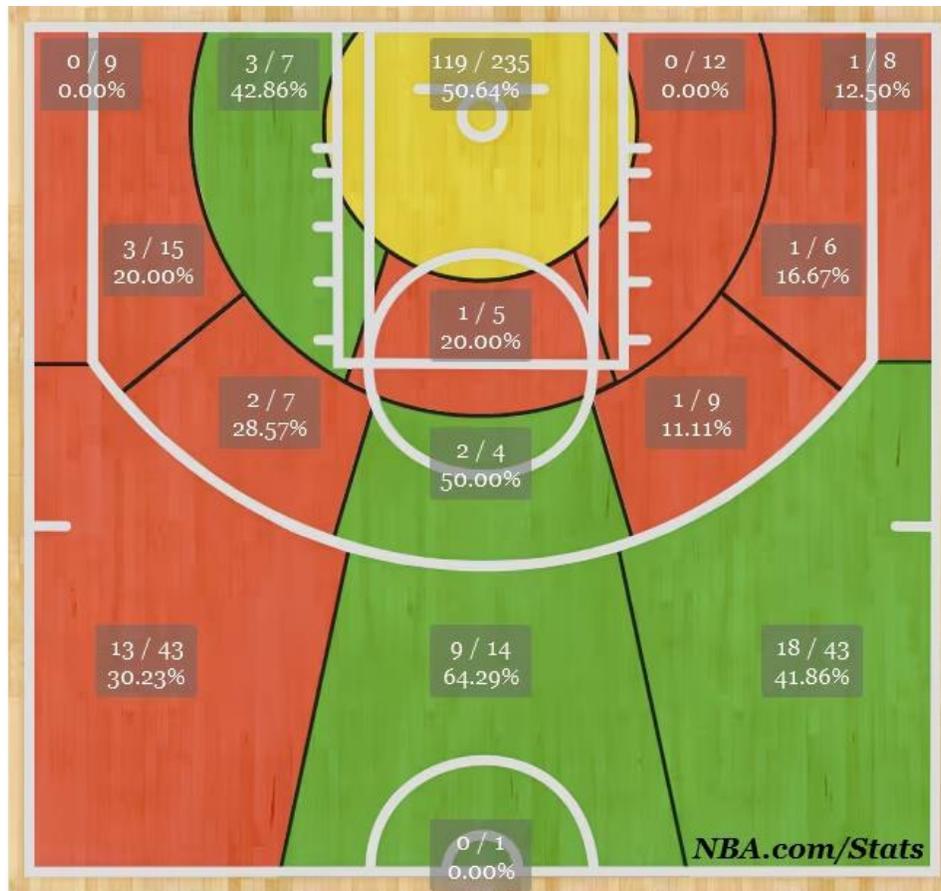


Figura 17: Visualização de zonas utilizada pela NBA

O conjunto de polígonos apresentado na figura 15 tem o objetivo de dividir a quadra levando em consideração a sua geometria e marcações. Isso é crucial para manter as divisões correspondentes a eventos distintos, como ocorre ao redor do garrafão, por exemplo. Essa segmentação auxilia na compreensão das regras e estratégias do basquete, uma vez que, como no exemplo anterior, as zonas ajudam a estabelecer uma separação natural dos locais de arremessos de dois e três pontos.

Além da separação que respeita os limites da quadra, são feitas outras divisões para entender melhor o uso dessas regiões. Isso permite àqueles que observam o gráfico compreender qual parte do garrafão foi mais utilizada, qual lado um time tende a atacar mais e outros fatores relevantes. Essas informações são cruciais para analisar a eficiência de uma equipe ou jogador, bem como para desenvolver estratégias de jogo.

Para começar o desenvolvimento dessa visualização, utilizando o conhecimento adquirido previamente, constatou-se que a representação de polígonos utilizando SVG seria mais eficiente em termos de desenvolvimento e geraria os

mesmos benefícios para o projeto quando comparada a outros métodos. Por esse motivo, direcionaram-se esforços para a construção dos polígonos no software Adobe Illustrator, com a finalidade de convertê-los em formato SVG para posterior inclusão no código. As figuras 16 e 17 buscam ilustrar como foi feito o desenho inicialmente e como ele foi aplicado na imagem da quadra.

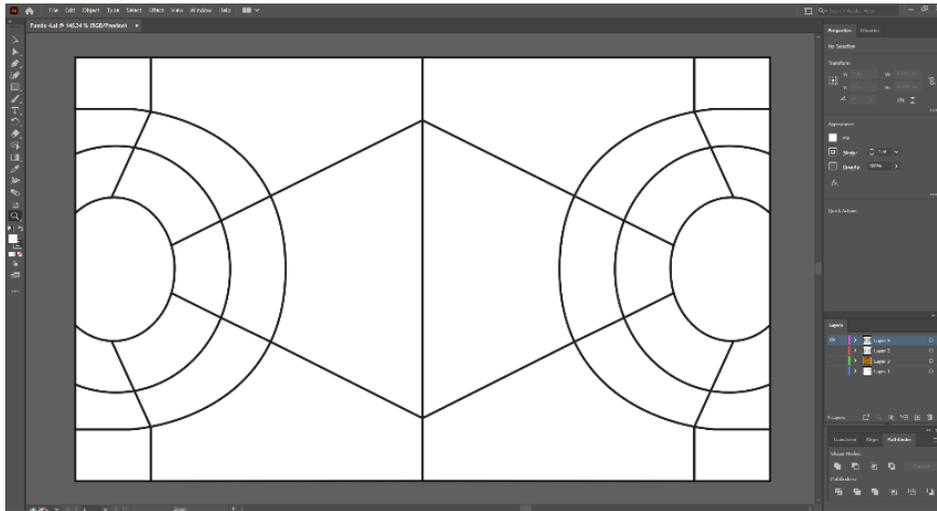


Figura 18: Polígonos desenhados no Illustrator

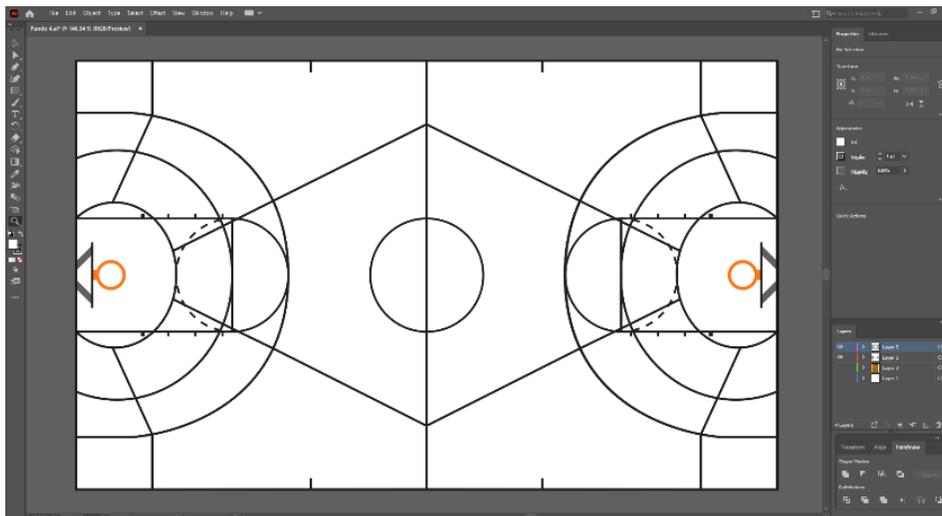


Figura 19: Polígonos desenhados no Illustrator aplicados na quadra

Estes polígonos são convertidos em SVG's, que consistem fundamentalmente em elementos HTML contendo um atributo "D". Este atributo representa o trajeto a ser traçado na imagem [12]. Dado que a aplicação requer apenas este atributo e que os polígonos possuem formas fixas, foi concebido um vetor de texto encarregado de armazenar o valor de 'D' para cada um dos polígonos acima.

Depois de desenhar os polígonos, foram mobilizados esforços para encontrar uma escala que ilustrasse bem a taxa de conversão em cada polígono. Chegou-se à conclusão de que seria apropriado adotar uma escala que transitasse do Vermelho, representando uma taxa de conversão baixa, até o Verde, simbolizando uma taxa de conversão alta, igual a escala utilizada no mapa de calor convencional. Por essa razão, optou-se por usar a mesma escala, que pode ser visualizada na figura 18.



Figura 20: Escala utilizada nos mapas de calor

Sempre que o usuário opta por essa visualização, essa escala é ao lado do mapa da quadra de basquete para ajudar o usuário a entender melhor as cores dos polígonos. A figura 19 ilustra como o usuário observa essa informação.

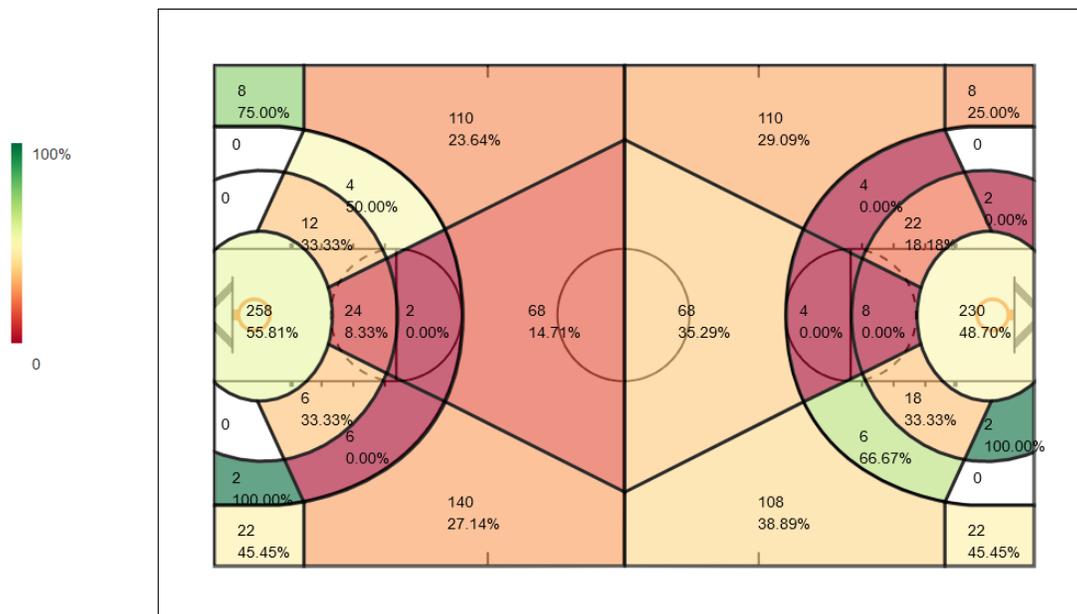


Figura 21: Exemplo de como o usuário visualiza a escala ao lado da quadra

Nesse cenário não existe nenhum tratamento a ser realizado previamente, então o componente do formulário envia os dados diretamente para o de visualização depois do espelhamento, parecido com o que é feito no caso da visualização de círculos.

Nesta situação, o componente de visualização tem a tarefa de traçar os caminhos armazenados como SVGs na forma de polígonos. Isso é necessário graças à estrutura deste tipo de dado que conta com um método particularmente útil para esse tipo de visualização. Esta conversão é realizada pelo método `pathDataToPolys`, proveniente da biblioteca `svg-path-to-polygons` [13]. Depois de desenhados, a

aplicação percorre os eventos e faz uso do método citado anteriormente *d3.polygonContains*, que é responsável por verificar se o polígono tem um ponto (X, Y) dentro de sua área. Fazendo o uso desse método, a aplicação consegue identificar a os eventos de cada polígono e preenchê-lo com uma cor que corresponde à taxa de sucesso de arremessos (acertos divididos/total de tentativas) que ocorreram dentro de suas fronteiras utilizando a mesma escala do mapa de calor convencional.

Além dos polígonos coloridos, essa visualização oferece uma caixa de informações quando um polígono é focado (por exemplo, quando o mouse passa por cima). Ela é responsável por exibir estatísticas detalhadas sobre os arremessos. Por fim, para reduzir um pouco a dependência das caixas de informações, o número total de arremessos e a porcentagem de acertos são exibidos como texto no centro do polígono.

As figuras a seguir objetivam exemplificar a visualização discutida acima. A figura 20 representa o cenário de um único jogo, enquanto a 20 figura engloba todos os arremessos de todos os jogos disponíveis da temporada de 2021. Em todas as imagens a seguir, o lado esquerdo representa os arremessos realizados pelo Flamengo, enquanto o lado direito mostra os arremessos dos adversários.

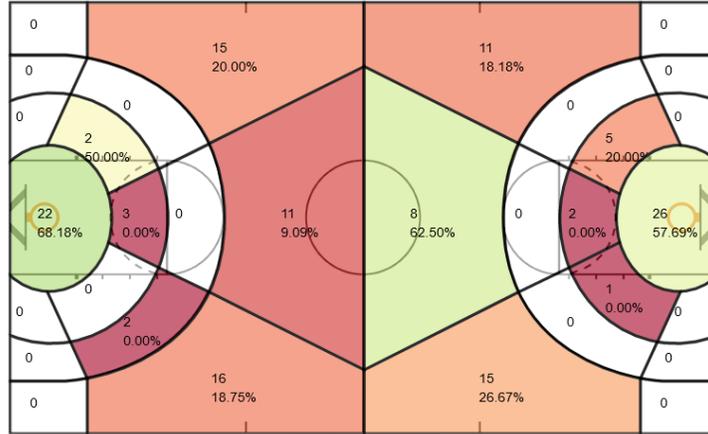


Figura 22: Exemplo de mapa de calor customizado

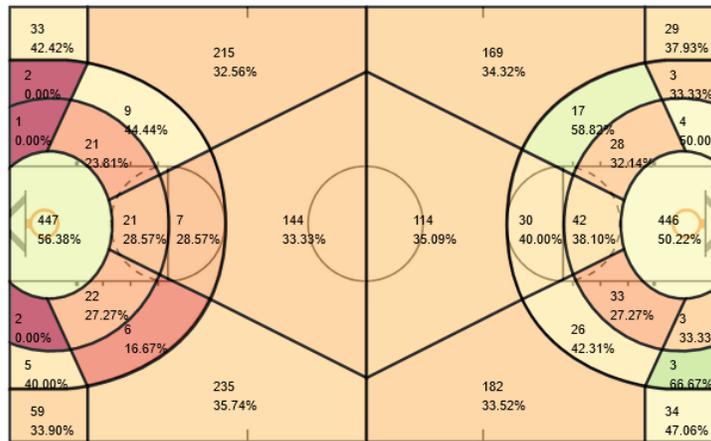


Figura 23: Mapa de calor customizado de todas as partidas da temporada de 2021

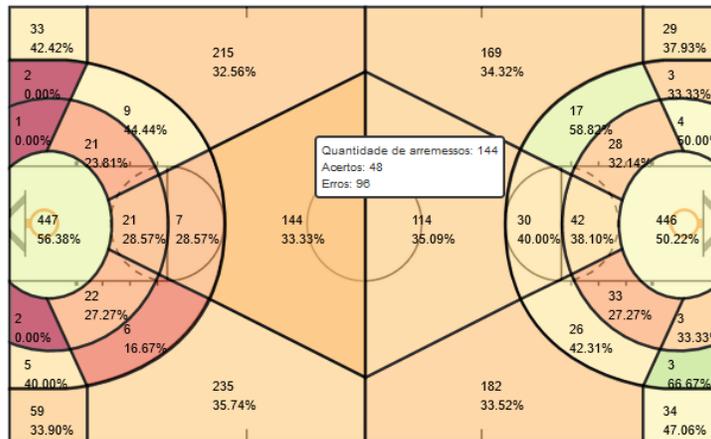


Figura 24: Caixa de informações do mapa de calor customizado de todas as partidas da temporada de 2021

Ao observar as figuras 20, 21 e 22, concluímos que essa abordagem ajudou a reduzir a "poluição" visual causada pela grande quantidade de retângulos coloridos.

Essa mudança gerou melhorias significativas na compreensão das áreas mais bem utilizadas na quadra. Ainda assim, a dificuldade em entender uma escala de cores não tão intuitiva ainda persiste.

Mapa de círculos agrupados

Diante das então presentes dificuldades, optamos por manter os mapas de calor da forma que foram implementados e testar o agrupamento de pontos sobrepostos na primeira visualização. Essa abordagem tem como o principal objetivo solucionar o problema encontrado na visualização tradicional de círculos, que era o grande ruído de dados. Durante estudos, descobrimos que agrupar os dados que causam confusão é uma prática comum na visualização, isso evidenciou a necessidade de testes com esse método. Os testes evidenciaram que essa abordagem representou uma grande melhora na redução do ruído de dados, mantendo uma precisão satisfatória na localização.

Antes de implementar a nova maneira de visualizar os dados, foi realizada uma análise intensiva para determinar as melhores maneiras de realizar esse agrupamento. Durante a fase de estudos, nos deparamos com a possibilidade de implementar um Algoritmo Guloso para detectar as colisões. Isso porque, de acordo com Rocha (2004) [14], na seção de seu trabalho intitulada por “Propriedade de escolha gulosa”, esse algoritmo tem como característica realizar a melhor escolha local em cada etapa, assim poderíamos facilmente testar contextos de agrupamento diferentes e garantir um funcionamento adequado da visualização.

Ao lidar com a detecção de colisões, estamos essencialmente tentando detectar interações entre diferentes círculos. Essas interações são dinâmicas e podem ser influenciadas por outras, já que elas alteram as posições e raios de cada círculo. O Algoritmo Guloso, com sua habilidade de realizar a melhor escolha possível a cada etapa com base nas condições atuais, torna-se uma boa alternativa para lidar com essa natureza dinâmica.

Especificamente, em um cenário em que os círculos estão se movendo, crescendo e se sobrepondo, o algoritmo escolhido pode identificar e resolver de maneira efetiva as sobreposições mais claras primeiro. Isto é alcançado por meio do foco na escolha localmente ótima em cada etapa, ou seja, resolver o cenário de colisão escolhido primeiro. Após a realização de testes em diferentes cenários, concluímos que nossa melhor escolha local seria resolver primeiro as colisões que ocorrem entre círculos com os menores raios da lista.

Em diversos cenários testados, constatamos que, ao detectar uma sobreposição localmente ótima, deveríamos excluir ambos os eventos da lista e criar um evento intermediário com o raio proporcional a quantidade de ocorrências. Além disso, era preciso posicionar este novo círculo em uma posição intermediária entre os dois.

Após averiguar a funcionalidade da implementação, percebeu-se a necessidade de um ajuste no deslocamento. Concluiu-se que precisávamos aplicar um peso nesse deslocamento e, depois de alguns testes, decidiu-se que a distância do evento novo deveria ser inversamente proporcional ao tamanho do raio dos eventos que colidiram. Esta decisão garante que círculos com muitos eventos mantenham sua precisão. Após uma série de testes, essa abordagem foi adotada como a mais adequada.

Após estabelecer a metodologia apropriada para o tratamento desses dados, foi alcançada a versão ideal para a execução desta estratégia. Este caso específico exige uma lógica um pouco mais complexa que outras abordagens. Para agrupar os dados, o programa inicializa uma variável "Feito" como falsa e entra em um ciclo que se prolonga até que "Feito" seja alterada para verdadeira. Dentro deste ciclo, são criados campos para armazenar o Raio mínimo e os dois eventos que colidiram e satisfizeram o critério de solução local ótima. Dentro do ciclo, a variável "Feito" é então definida como verdadeira e, posteriormente, a iteração ocorre para cada par de eventos distintos na lista. Caso sejam eventos diferentes, porém do mesmo tipo (por exemplo, um Arremesso), verifica-se se ambos ou nenhum dos eventos pertencem ao Flamengo. Se a condição for satisfeita, é calculada a distância entre os dois eventos.

A sobreposição é determinada calculando a distância euclidiana entre as coordenadas dos dois eventos e comparando-a com os raios dos eventos. Considerando que os pontos possuem uma posição X e Y variando de 0 a 100 na API, e que a quadra tem dimensões distintas de altura e largura, é necessário utilizar uma escala linear para organizar esses pontos no mapa de maneira correta. Para assegurar que os círculos também estão colidindo visualmente, a mesma escala é utilizada nesse cálculo.

Caso a distância seja menor do que a soma dos dois raios, o algoritmo verifica se essa sobreposição é considerada localmente ótima, comparando o raio dos dois

eventos com o menor raio registrado até o momento atual. Se for, a aplicação registra esses eventos como a solução localmente ótima e altera a variável "Feito" para "falso", garantindo a verificação de outras possíveis colisões. Quando todas as colisões forem resolvidas, a variável "Feito" não será mais alterada para "Falso", terminando assim o ciclo.

Após terminar de verificar todos os pares de eventos, a aplicação cria um novo evento cujas propriedades são uma combinação das propriedades dos dois eventos que se sobrepunham. Este novo evento recebe o mesmo tipo de evento e equipe que os dois eventos que colidiram, e o seu raio é alterado para a raiz quadrada do número de eventos que colidiram vezes o raio mínimo, que tem valor 6 neste caso. Posteriormente, os eventos que se sobrepueram são removidos da lista de eventos e o novo evento é adicionado. O ciclo continua até que não existam mais sobreposições entre os eventos. Após isso, o vetor de eventos é ordenado em ordem decrescente de acordo com o raio dos eventos, garantindo que os eventos menores não serão sobrepostos por maiores na visualização

Por fim, a lista de eventos sem superposições é enviada para o componente de visualização. Ao receber os dados, este componente cria uma visualização de pontos na qual a posição, o raio e a cor dos pontos são determinados com base nos dados recebidos. Além disso, a aplicação adiciona interatividade, permitindo que, ao posicionar o ponteiro do mouse sobre um círculo, seja exibida uma caixa de informações complementares ("Tooltip").

As figuras 23, 24 e 25 buscam representar essa abordagem de visualização dos eventos. Enfatiza-se que, em todas as representações gráficas a seguir, o lado esquerdo ilustra os arremessos realizados pela equipe do Flamengo, enquanto o lado direito exibe os arremessos realizados pelas equipes adversárias.

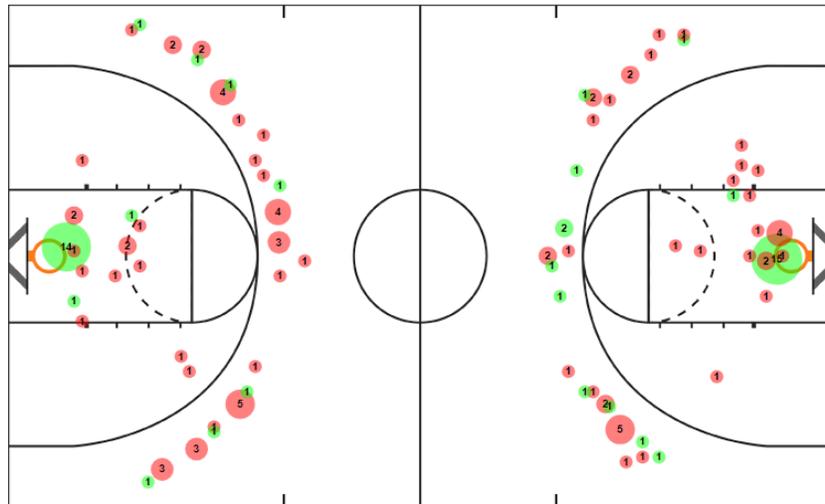


Figura 25: Mapa de círculos agrupados

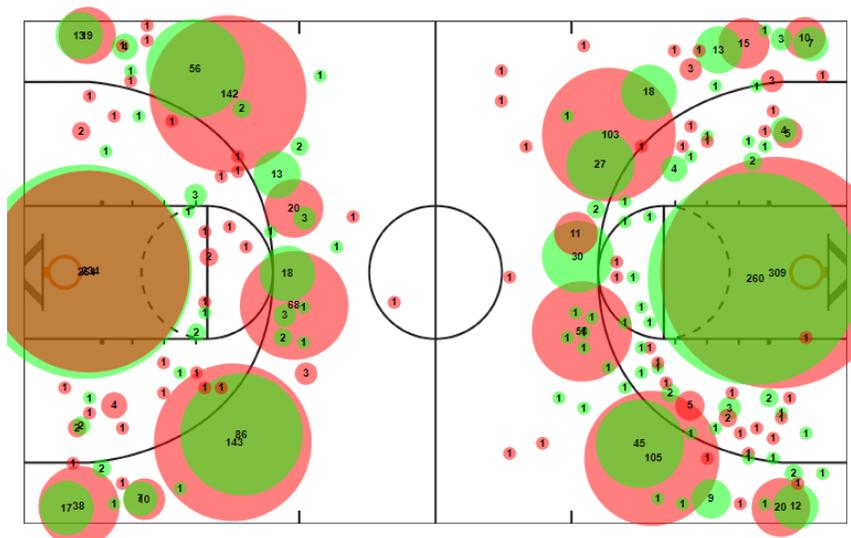


Figura 26: Mapa de círculos agrupados para todas as partidas da temporada de 2021

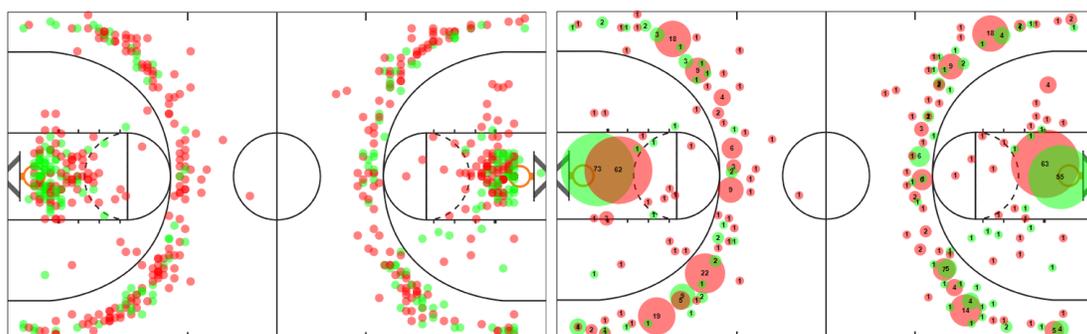


Figura 27: Comparação entre a exibição sem agrupamento (à esquerda) e com agrupamento (à direita) para as partidas FRA x FLA (09/06/2022), FLA x FRA

(04/06/2022), FLA x FRA (02/06/2022), FRA x FLA (28/05/2022) e MIN x FLA (17/05/2022)

As figuras 23, 24 e 25 buscam evidenciar que este tipo de visualização oferece uma certa precisão na localização dos eventos, e mitiga os problemas enfrentados na visualização comum de pontos por meio da sobreposição de ponto. Essa abordagem possibilita um entendimento fácil do tipo de evento (acerto/erro) e da quantidade de eventos agrupados. Além disso, a simplicidade visual dessa visualização contribui para uma melhor compreensão das áreas pouco utilizadas. No entanto, é importante ressaltar que a interpretação da área do círculo depende da informação textual que evidencia a quantidade de arremessos. Esta interpretação pode se tornar menos clara em situações em que os círculos têm centros próximos um do outro, dificultando a identificação do texto.

Visão geral

Todas as visualizações que foram testadas e implementadas trouxeram características bem únicas, se mostrando úteis em diversos cenários diferentes, mas nota-se que a visualização proposta de pontos sem sobreposição trouxe um conjunto muito forte de clareza, limpeza e precisão, evidenciando que é uma proposta de visualização mais adequada para o entendimento geral de uma partida.

Sistema

O sistema, apesar de complexo em sua implementação, oferece uma utilização simples e direta. O usuário acessa a URI da aplicação, preenche o formulário de acordo com os dados que deseja visualizar, envia o formulário e utiliza as ferramentas de visualização disponíveis. O mapa da quadra de basquete exibe todos os dados e permite a interação através do zoom e da caixa de informações para eventos específicos.

Testes

Ao longo do desenvolvimento e da pesquisa, foram conduzidas inspeções da arquitetura e do código, visando possíveis extensões ou refatorações do sistema criado ao longo do desenvolvimento do projeto. Também foram realizadas comparações de esforço de desenvolvimento entre muitas alternativas ao longo do desenvolvimento. Para finalizar, estudos de desempenho foram conduzidos para avaliar o desempenho do sistema com o objetivo principal de evitar lentidões causadas por requisições com grande quantidade de eventos.

Além disso, foram realizadas inspeções da interface com o usuário, a fim de avaliar e melhorar sua usabilidade, além de identificar pontos fortes e fracos nos diferentes tipos de visualizações implementadas e testadas ao longo do projeto.

Todos os testes trouxeram muitos benefícios, como identificar e implementar otimizações, direcionar o estudo de visualizações, otimizar o esforço de desenvolvimento, além da análise e melhoria de desempenho do sistema. No entanto, é importante considerar que esses testes também possuem suas limitações em relação à cobertura completa das necessidades e percepções dos usuários.

Implementação e avaliação

Planejamento e execução de testes funcionais

O projeto descrito pode ser categorizado em duas seções principais para simplificar nossos procedimentos de teste. A primeira seção concentra-se em garantir o funcionamento adequado da aplicação, especificamente, a confirmação de que o formulário opera conforme o esperado e realiza as solicitações apropriadas para a API. Em adição, um teste meticuloso de todos os métodos de visualização de dados é imperativo.

Para assegurar a funcionalidade do formulário, foram realizados testes através de preenchimentos diversos e todas as solicitações resultantes foram comparadas com aquelas feitas por meio do Postman. Postman é uma plataforma direcionada aos desenvolvedores para a criação, construção, teste e iteração de suas APIs (Postman, 2021). Esses testes revelaram que as respostas recebidas pela aplicação eram idênticas às obtidas através do Postman, garantindo assim, a correta operação do formulário. As figuras 26 e 27 buscam ilustrar o resultado de uma requisição feita pelo envio do formulário e outra realizada pelo Postman.


```

GET https://apibird.tecgraf.puc-rio.br/v1/events/1/24347?page=1&limit=500&event_types=A2E, A2C, A3E, A3C

Params • Authorization • Headers (9) Body Pre-request Script Tests Settings

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON ↕

1
2 "page": 1,
3 "size": 139,
4 "total": 139,
5 "data": [
6   {
7     "league_id": 1,
8     "league_name": "NBB",
9     "match_id": 24347,
10    "event_id": 7,
11    "season": 2021,
12    "code": "A3C",
13    "time": {
14      "quarter": {
15        "id": 1,
16        "clock": "00:09:52",
17        "elapsed_seconds": 8,
18        "remaining_seconds": 592
19      },
20      "elapsed_seconds": 8,
21      "date": "2022-06-09T18:30:37"
22    },
23    "score": {
24      "home": 3,
25      "away": 0
26    },
27    "player": {
28      "id": 26,
29      "name": "Jhonatan Luz dos Santos",
30      "nickname": "Jhonatan",
31      "number": 7
32    },
33    "team": {
34      "id": 14,
35      "name": "Franca",
36      "acronym": "FRA"
37    },
38    "position": {
39      "x": 77.0,

```

Figura 29: Requisição feita pelo postman

Com base nas figuras 26 e 27 apresentadas, observa-se que ambas as solicitações retornaram um total de 139 elementos. Além disso, o primeiro elemento em ambas as listas representa o evento "Arremesso de três pontos convertido", executado pelo jogador "Jhonatan Luz dos Santos". Além de conferir o primeiro elemento, ambos os vetores foram comparados manualmente e concluímos que ambas as solicitações resultaram na mesma lista de eventos.

A segunda seção do projeto exigiu um exame um pouco mais detalhado dos métodos de visualização. O primeiro passo envolveu solicitações para um único jogo com o objetivo de verificar se a exibição estava devidamente mapeada e se o tamanho do campo correspondia às dimensões originais. A preocupação inicial era verificar a conformidade do garrafão com as medidas do campo da NBB. As figuras 28 e 29 comprovam a precisão do mapa exibido. O evento "Arremesso de dois pontos

convertido" aparece pouco antes da linha do garrafão e o "Arremesso de três pontos convertido" aparece logo após, indicando que o tamanho do garrafão está conforme o ideal.

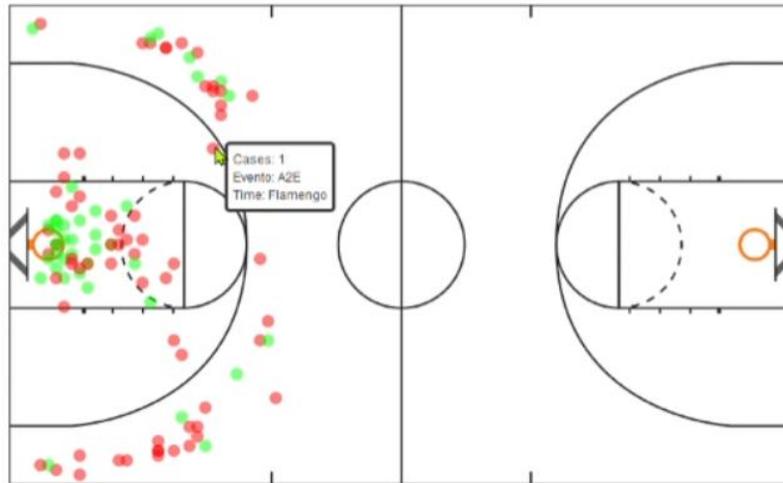


Figura 30: Arremesso de dois pontos errado no limite do garrafão indicado pelo mouse

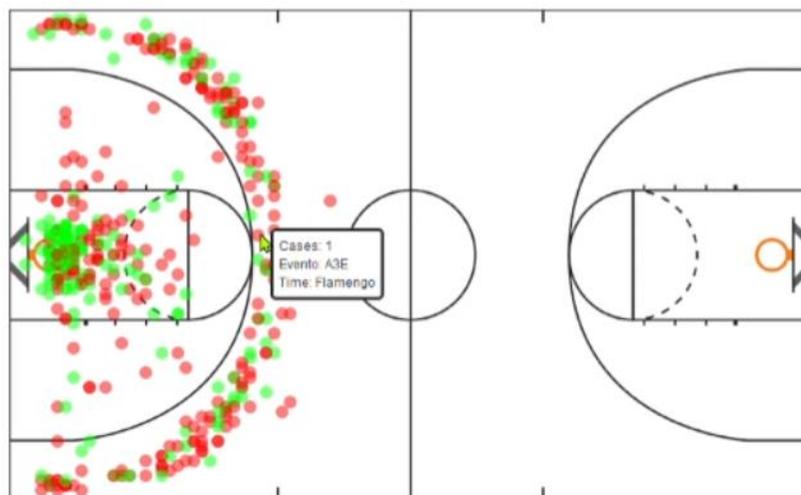


Figura 31: Arremesso de três pontos errado logo após o limite do garrafão indicado pelo mouse

Posteriormente, para dar continuidade aos testes, procuramos por um ponto específico na solicitação feita pelo Postman e buscamos sua localização nas coordenadas previstas. Após a visualização desse ponto, confirmamos a exatidão da visualização.

A próxima etapa consiste em conduzir testes para garantir que todas as visualizações estejam apresentando os dados de maneira precisa e correta. Para iniciar esses testes, optou-se por um escopo inicial mais restrito: foram exibidos os

arremessos de somente quatro jogadores do Flamengo na Visualização de círculos.

Os resultados obtidos foram então comparados com todas as outras formas de visualização para validar o seu funcionamento. As figuras 30, 31, 32 e 33 foram fornecidas para ilustrar esses testes.

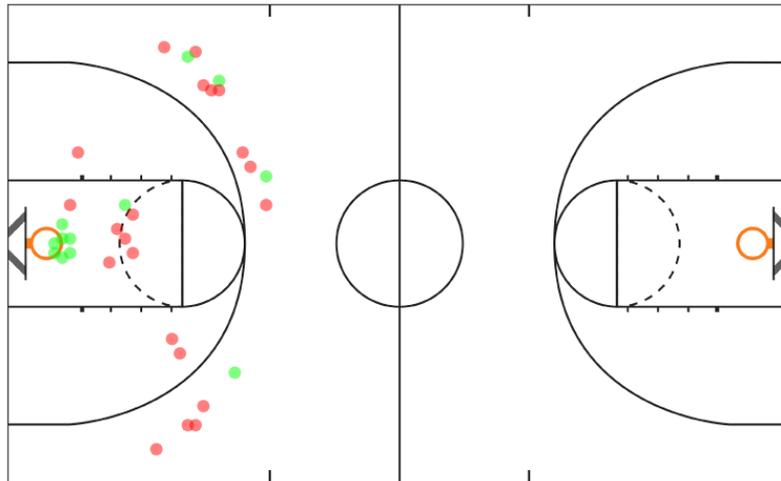


Figura 32: Visualização de círculos dos arremessos dos jogadores Rafael, Carlos, Brandon e Tulio na partida de 09/06/2022

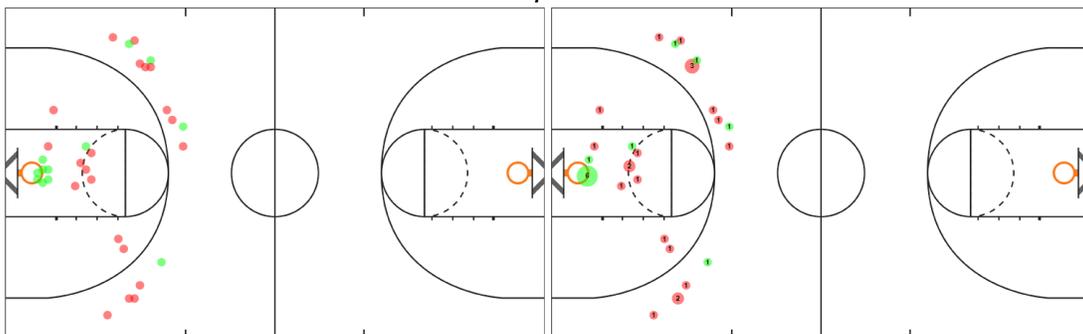


Figura 33: Comparação da Visualização de círculos (à esquerda) com a Visualização agrupada (à direita) dos arremessos dos jogadores Rafael, Carlos, Brandon e Tulio na partida de 09/06/2022

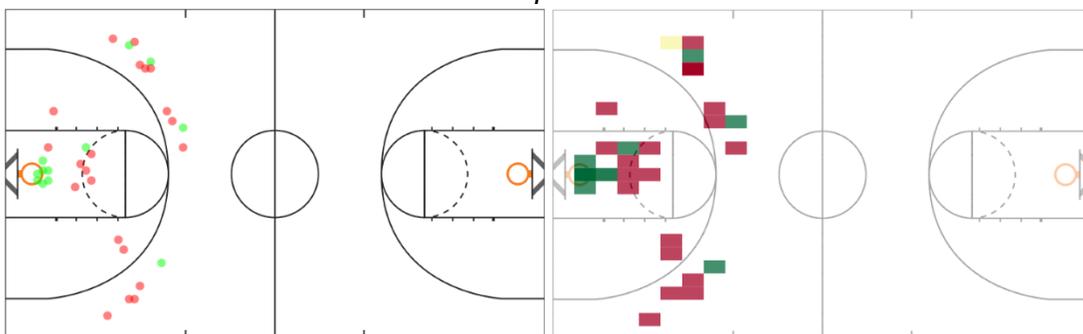


Figura 34: Comparação da Visualização de círculos (à esquerda) com o Mapa de calor (à direita) dos arremessos dos jogadores Rafael, Carlos, Brandon e Tulio na partida de 09/06/2022

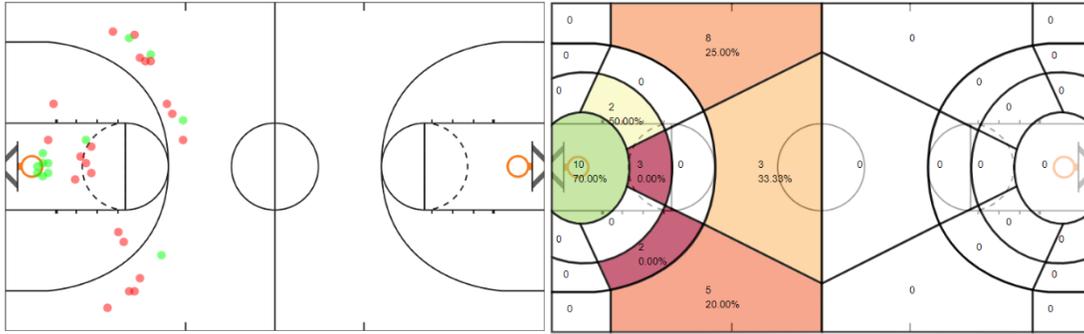


Figura 35: Comparação da Visualização de círculos (à esquerda) com o Mapa de calor customizado (à direita) dos arremessos dos jogadores Rafael, Carlos, Brandon e Tulio na partida de 09/06/2022

Com base nas figuras apresentadas, conclui-se que, nesse cenário específico, todas as visualizações estão consistentes e exibem os mesmos dados de maneira diferente. Uma vez que todos os algoritmos de visualização foram desenvolvidos de maneira genérica, a funcionalidade verificada acima é aplicável a conjuntos de dados de maior tamanho. Foram realizados mais testes com todas as partidas da temporada, no entanto, para facilitar a compreensão, as imagens apresentam uma quantidade de dados menor.

Em conclusão, foram conduzidos diversos testes para identificar a solução local ótima empregando o Algoritmo Guloso na Visualização de círculos agrupados. Ao longo desses experimentos, identificamos que dois critérios de agrupamento proporcionavam resultados notáveis: maior sobreposição e menor raio.

A figura 34 procura ilustrar a diferença observada. Ela exibe o critério do menor raio à esquerda e o da maior sobreposição à direita. Os jogos retratados nas imagens são: FRA x FLA (09/06/2022), FLA x FRA (04/06/2022), FLA x FRA (02/06/2022), FRA x FLA (28/05/2022) e MIN x FLA (17/05/2022).

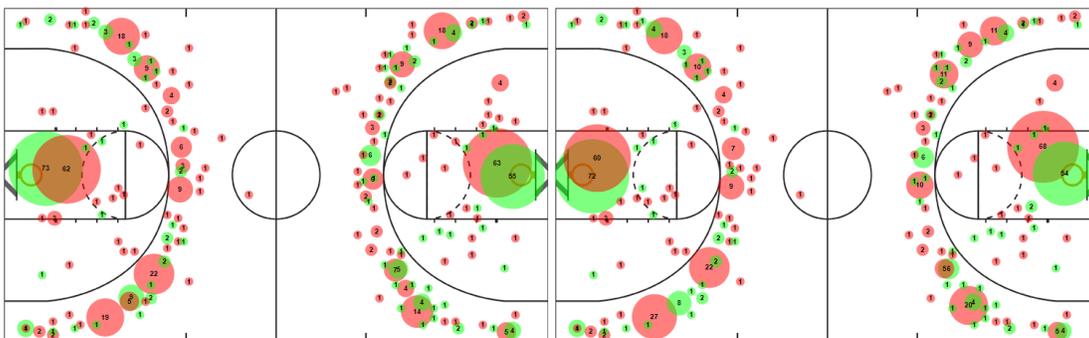


Figura 36: Comparação entre o critério de menor raio (à esquerda) e da maior sobreposição (à direita) das partidas FRA x FLA (09/06/2022), FLA x FRA

(04/06/2022), FLA x FRA (02/06/2022), FRA x FLA (28/05/2022) e MIN x FLA (17/05/2022)

Depois de testar ambas as opções, chegamos à conclusão de que o critério do menor raio apresentava uma representação mais precisa da distribuição de arremessos durante um jogo. Portanto, esse critério foi o escolhido para ser aplicado na visualização.

Planejamento e execução de outros testes

Após a validação do funcionamento da aplicação, realizamos alguns exercícios com o objetivo de identificar possíveis melhorias na intuitividade do site. A finalidade principal de todas as comparações e testes realizados foi facilitar o preenchimento do formulário e identificar áreas para aprimoramento na visualização dos dados. Como resultado desses testes, obtivemos avanços significativos, como a implementação do preenchimento automático das opções dos campos do formulário com dados da API, substituindo a necessidade de inserção manual dos IDs das partidas, por exemplo. Adicionalmente, implementamos caixas de informações exibidas quando o usuário passa o mouse por cima do evento nas visualizações para fornecer informações adicionais que inicialmente não eram exibidas.

Comentários sobre a implementação

Ao longo do desenvolvimento do projeto, diversas dificuldades associadas ao estudo e implementação de algoritmos de visualização foram encontradas. Durante esse período, consideráveis esforços foram direcionados para garantir o funcionamento adequado de todos os métodos, bem como sua performance satisfatória.

Em um estágio avançado do projeto, identificamos uma inconsistência na Visualização de Círculos com Agrupamento. Em resposta a isso, decidimos alterar a lógica de agrupamento para empregar o Algoritmo Guloso. Esta alteração corrigiu as inconsistências que havíamos encontrado, mas resultou em uma ligeira diminuição de performance. Como a troca de algoritmo foi feita perto do final do projeto, não houve tempo de otimizar e focar na performance do novo.

Considerações finais

Contribuições

Ao estabelecer um comparativo entre as visualizações desenvolvidas neste projeto e os métodos convencionalmente utilizados para a visualização de eventos de

uma partida de basquete, torna-se evidente a notável melhoria em termos de intuitividade e clareza na apresentação dos eventos. A aplicação proporciona a visualização de arremessos através de quatro métodos distintos, cada um com suas características e benefícios específicos. Esta diversidade de opções amplia as ferramentas disponíveis para os profissionais da área, auxiliando na tomada de decisões.

Além disso, a aplicação expande a sua funcionalidade para além da visualização de arremessos, oferecendo a possibilidade de exibir outros eventos, tais como violações e erros, através da Visualização de círculos, que podem ser apresentados de forma agrupada ou não. Mas é importante enfatizar que as visualizações de Mapa de calor só foram pensadas para arremessos, uma vez que a escala é feita pela taxa de conversão.

Cabe ressaltar que, apesar de alguns eventos ainda não possuírem coordenadas registradas na API consultada pela aplicação, o projeto foi desenvolvido de forma a suportar qualquer tipo de evento com posições, sejam eles já existentes ou que venham a ser adicionados futuramente.

Aprendizado

Este projeto representou uma valiosa oportunidade para o aprofundamento do meu conhecimento acerca do desenvolvimento de software, APIs, algoritmos de visualização e diversas formas de apresentação de conteúdo similar. Desde a fase inicial, o projeto se mostrou um grande desafio, exigindo o aprimoramento da minha competência em desenvolvimento Front-end para começar o desenvolvimento de uma aplicação Angular e selecionar as bibliotecas apropriadas para as visualizações planejadas e possíveis.

Além disso, a necessidade de investigar alternativas de visualizações contribuiu para um significativo acúmulo de conhecimento sobre como gerir dificuldades ocasionadas pelo grande volume de dados, como, por exemplo, o problema de ruído de dados. Ademais, o projeto proporcionou uma oportunidade valiosa para investigar e compreender a aplicação de métodos amplamente utilizados na atualidade, como os mapas de calor. A oportunidade de implementar tais métodos ofereceu insights profundos sobre a lógica subjacente que orienta essas técnicas.

Em última análise, o projeto reiterou a importância do planejamento criterioso de um sistema antes do início do desenvolvimento. O processo de desenvolvimento tornou-se consideravelmente mais simples após a elaboração dos diagramas

apresentados anteriormente neste relatório. A compreensão e o esclarecimento do funcionamento da aplicação facilitaram o desenvolvimento consistente dela, minimizando a necessidade de mudanças ao longo do tempo.

Se houvesse a possibilidade de refazer o projeto, certamente optaríamos pelo uso do Algoritmo Guloso desde o início, realizando otimizações ao longo do tempo para assegurar tanto a consistência dos dados quanto a boa performance. Ademais, adotaria uma abordagem mais criteriosa na segmentação dos componentes que constituem o site. Embora a estruturação atual seja adequada ao atual problema, uma divisão mais bem planejada permitiria uma maior adaptabilidade do projeto, preparando-o para futuras expansões, como a integração de eventos em um campo de Futebol, por exemplo.

Projetos Futuros

Este projeto expandiu os meus horizontes, introduzindo-me a uma variedade de conceitos e práticas diretamente ligados à implementação e visualização de dados. Acredito que ainda existe uma vasta margem para estudo adicional e aprofundamento dentro deste campo. Além disso, acredito que ainda há um número significativo de áreas que carecem de visualizações que sejam ao mesmo tempo intuitivas e claras. Este fato evidencia que a área representa um campo promissor para futuros estudos acadêmicos.

Referências Bibliográficas

- [1] Infogram. “O que é visualização de dados?”. Disponível em <https://infogram.com/pt/pagina/visualizacao-de-dados>. Acesso em: 07 out. 2022.
- [2] Carvalho, D. R; Dallagassa, M. R. Mineração de dados: aplicações, ferramentas, tipos de aprendizado e outros subtemas. Disponível em: <https://revistas.ufpr.br/atoz/article/view/41340/25332>. Acesso em: 21 jun. 2023
- [3] SCIEPUB. Big Data and Visualization: Methods, Challenges and Technology Progress. Disponível em <http://pubs.sciepub.com/dt/1/1/7/#Reference12>. Acesso em: 07 out. 2022.
- [4] ANGULAR. Angular. Disponível em <https://angular.io/> . Acesso em: 04 out. 2022.
- [5] ANGULAR. Google Cloud. Disponível em <https://www.madewithangular.com/sites/google-cloud>. Acesso em: 04 out. 2022.
- [6] ANGULAR. Mc Donalds. Disponível em <https://www.madewithangular.com/sites/mcdonalds> . Acesso em: 04 out. 2022.
- [7] ANGULAR. Apple Support. Disponível em <https://www.madewithangular.com/sites/apple-support> . Acesso em: 04 out. 2022.
- [8] ANGULAR. Forbes. Disponível em <https://www.madewithangular.com/sites/forbes>. Acesso em: 04 out. 2022.
- [9] Adobe. Arquivos SVG. Disponível em: <https://www.adobe.com/br/creativecloud/file-types/image/vector/svg-file.html#:~:text=SVG%3A%20perguntas%20frequentes-.O%20que%20%C3%A9%20um%20arquivo%20SVG%3F,e%20linhas%20em%20uma%20grade> . Acesso em: 21 jun. 2023.
- [10] Nature Maps (2023). Heat maps are useful for visualizing multivariate data but must be applied properly. Disponível em: <https://www.nature.com/articles/nmeth.1902>. Acesso em: 20 jun. 2023.
- [11] NBA (2023). Bucks Features: Shot Charts. Disponível em: <https://www.nba.com/bucks/features/boeder-shot-charts>. Acesso em: 20 jun. 2023.
- [12] Mozilla. D. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/d> . Acesso em: 21 jun. 2023.
- [13] npm. Svg-path-to-polygons. Disponível em: <https://www.npmjs.com/package/svg-path-to-polygons> . Acesso em: 21 jun. 2023
- [14] Rocha, H. V. (2004). Algoritmos Gulosos. Instituto de Computação – UNICAMP. Disponível em:

<https://www.ic.unicamp.br/~rocha/msc/complex/algoritmosGulososFinal.pdf>. Acesso em: 21 jun. 2023.

Apêndices

Manual de usuário

Introdução

Este manual tem como objetivo auxiliar e guiar a utilização da aplicação disponível no repositório do Github com link <https://github.com/abreusleo/FinalProject>. Além desse documento, não deixe de consultar o README do projeto.

Visão geral

Para utilizar a aplicação, o usuário só precisa preencher um formulário e esperar até que os dados processados sejam exibidos na imagem da quadra de basquete.

É importante enfatizar que algumas visualizações são próprias para Arremessos, por esse motivo, não será possível observar qualquer evento que não seja arremesso nos mapas de calor. Isso ocorre porque a escala de cores é definida pela taxa de conversão.

Requisitos do sistema

Para rodar a aplicação, é necessário possuir os seguintes softwares:

- Nodejs
- Angular

Instalação e execução

Para instalar a aplicação, o usuário precisa acessar o link do Github <https://github.com/abreusleo/FinalProject> e clonar o repositório para a sua máquina. Depois de clonado, o usuário precisa abrir a pasta do repositório no Prompt de comando e executar o comando "ng serve"

Primeiros passos

Para começar a utilizar a nossa aplicação, é muito simples. Tudo o que você precisa fazer é preencher o campo "Partidas" no formulário. Não se preocupe, você pode selecionar mais de uma partida, basta segurar a tecla Ctrl e clicar nas opções desejadas.

Após selecionar as partidas, você terá a liberdade de realizar diferentes combinações de escolhas. Basta prosseguir com o preenchimento do formulário e enviá-lo.

Aguarde um momento até que os dados sejam exibidos na quadra ou até que a imagem de carregamento desapareça, caso haja eventos que não tenham ocorrido ao longo da partida.

Agora que os dados estão visíveis na quadra, você tem total liberdade para explorá-los da maneira que preferir. Você pode aplicar zoom, aproveitar o campo de texto que aparece ao posicionar o mouse sobre um evento ou uma área e realizar qualquer análise que desejar.