



Trabalho de Conclusão de Curso

Realidade estendida para visualização de dados: Desenvolvimento de uma aplicação web imersiva com A-Frame

de Lilian Giselly Pereira Santos

orientado por

Prof. Dr. Ícaro Bezerra Queiroz de Araújo

Universidade Federal de Alagoas
Instituto de Computação
Maceió, Alagoas
27 de Novembro de 2024

UNIVERSIDADE FEDERAL DE ALAGOAS
Instituto de Computação

**REALIDADE ESTENDIDA PARA VISUALIZAÇÃO DE
DADOS: DESENVOLVIMENTO DE UMA APLICAÇÃO
WEB IMERSIVA COM A-FRAME**

Trabalho de Conclusão de Curso submetido
ao Instituto de Computação da Universidade
Federal de Alagoas como requisito parcial
para a obtenção do grau de Engenheiro de
Computação.

Lilian Giselly Pereira Santos

Orientador: Prof. Dr. Ícaro Bezerra Queiroz de Araújo

Banca Avaliadora:

Prof. Dr. Glauber Rodrigues Leite	IC-UFAL
Prof. Dr. Tácito Trindade de Araújo Tiburtino Neves	DCX-UFPB
Prof. Dr. Allan de Medeiros Martins	DEE-UFRN

Maceió, Alagoas
27 de Novembro de 2024

Catálogo na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

S237r Santos, Lilian Giselly Pereira.

Realidade estendida para visualização de dados : desenvolvimento de uma aplicação web imersiva com A-Frame / Lilian Giselly Pereira Santos. – 2024.

59 f. : il.

Orientador: Ícaro Bezerra Queiroz de Araújo.

Monografia (Trabalho de conclusão de curso em Engenharia de Computação) - Universidade Federal de Alagoas, Instituto de Computação. Maceió, 2024.

Bibliografia: f. 58-59.

1. Realidade estendida. 2. Visualização de dados. 3. A-frame (Aplicações web). I. Título.

CDU: 004.358



Trabalho de Conclusão de Curso - TCC

Formulário de Avaliação

Nome do Aluno Lilian Giselly Pereira Santos	
Nº de Matrícula 19111115	
Título do TCC (Tema) Realidade estendida para visualização de dados: Desenvolvimento de uma aplicação web imersiva com A-Frame	
Banca Examinadora	
Prof. Dr. Ícaro Bezerra Queiroz de Araújo Orientador	 Documento assinado digitalmente ICARO BEZERRA QUEIROZ DE ARAUJO Data: 27/11/2024 18:07:13-0300 Verifique em https://validar.iti.gov.br
Prof. Dr. Glauber Rodrigues Leite Membro da banca	 Documento assinado digitalmente GLAUBER RODRIGUES LEITE Data: 28/11/2024 08:52:13-0300 Verifique em https://validar.iti.gov.br
Prof. Dr. Tácito Trindade de Araújo Tiburtino Neves Membro da banca	 Documento assinado digitalmente TACITO TRINDADE DE ARAUJO TIBURTINO NEV Data: 28/11/2024 16:20:37-0300 Verifique em https://validar.iti.gov.br
Prof. Dr. Allan de Medeiros Martins Membro da banca	 Documento assinado digitalmente ALLAN DE MEDEIROS MARTINS Data: 02/12/2024 11:06:25-0300 Verifique em https://validar.iti.gov.br
Data da Defesa 27/11/2024	Nota Obtida 10,0 (Dez)
Observações 	
Coordenador do Curso De Acordo	
 Documento assinado digitalmente JOBSON DE ARAUJO NASCIMENTO Data: 02/12/2024 14:45:03-0300 Verifique em https://validar.iti.gov.br	
Assinatura	

Dedicatória

À todos que, com amor e paciência, me ajudaram a chegar até aqui.

Agradecimentos

Em primeiro lugar, agradeço aos meus pais, Elislaine e Wharley, por nunca terem medido esforços para que eu tivesse o melhor, sempre colocando minha educação e meu futuro em primeiro lugar. Cada esforço e sacrifício de vocês foi essencial para que eu chegasse até aqui. Carrego comigo tudo que aprendi com vocês, e espero poder retribuir com orgulho, em cada escolha e conquista.

Ao meu namorado, Pedro, companheiro de jornada e de tantos aprendizados: obrigada por estar sempre ao meu lado — me incentivando, ensinando e inspirando. Você me motiva a ser melhor todos os dias. De todos os presentes que a UFAL me deu, você é, sem dúvida, o mais especial.

Agradeço também à minha irmã Luanna, por ter sido lar longe de casa. E, junto com Thayná e Thauany, dividirem comigo não só o teto, mas o cuidado diário. À Laura, Laís, Larissa e Letícia, por me fazerem lembrar, mesmo de longe, das raízes e do afeto que me sustentam. Aos meus avós, Maria, Divá, Gelva e Lico pelo apoio e amor incondicional - sou imensamente grata por ter vocês na minha vida. À toda a minha família, obrigada por cada gesto de incentivo, carinho e suporte ao longo dessa jornada.

Sou imensamente grata às minhas amigas - em especial, Laura, Duda e Vitória, por serem meu ponto de refúgio mesmo há quilômetros de distância. E também à Eva, Bruna, Jô, Débora, Sophie, Duda, Laura, Paula e Paloma, por tantos momentos especiais. Aos colegas de turma, que caminharam ao meu lado, especialmente Sophia, Bruno e Fernanda, pelo apoio e amizade que fizeram toda a diferença. Ao projeto de extensão Katie, que me abriu portas para o crescimento pessoal e profissional e me apresentou amigas e colegas de profissão excepcionais. Em especial, Lívia, Rebeca, Ester e Natália.

Ao meu orientador, Ícaro Araújo, por todos os ensinamentos, pela confiança, pelas oportunidades e pela amizade. Sou profundamente grata por acreditar em mim quando eu mesma duvidei. Aos professores e membros da banca Allan Medeiros, Glauber Rodrigues e Tácito Neves, minha eterna gratidão por cada ensinamento e por contribuírem tanto para minha jornada acadêmica e pessoal. E a todos os professores que fizeram parte dessa trajetória, meu sincero agradecimento por cada aula e orientação longo do caminho.

E, por fim, à UFAL e aos laboratórios EASY, EDGE e NEES, por todas as experiências que me proporcionaram — pelos aprendizados, desafios, trocas e por contribuírem de forma tão significativa para minha formação e para os caminhos que virão.

Resumo

O avanço das tecnologias de realidade estendida (XR), que englobam a realidade virtual (VR) e aumentada (AR), tem ampliado as possibilidades de visualização de dados, permitindo experiências imersivas e interativas que tornam a análise de informações complexas mais intuitiva e acessível. Este trabalho apresenta o desenvolvimento de uma aplicação web imersiva utilizando o framework A-Frame, voltada para dispositivos compatíveis com XR. O projeto incluiu a implementação de gráficos interativos, como barras, linhas, dispersão, donuts e pizza, permitindo representar dados complexos em múltiplas dimensões e com alto grau de interatividade.

A metodologia adotada no projeto baseou-se na utilização da arquitetura Entity-Component-System (ECS) para modularizar e organizar o código, além de empregar métodos de normalização de dados para garantir a uniformidade na visualização. Foram desenvolvidos componentes customizados e sistemas integrados que facilitaram a criação de gráficos e a interação do usuário com os dados, como movimentar, redimensionar, rotacionar e filtrar informações diretamente no ambiente imersivo. Esses elementos foram projetados para serem renderizados de forma eficiente no navegador, ampliando a acessibilidade e eliminando a necessidade de software dedicado ou configurações complexas.

Os resultados demonstraram a viabilidade de criar gráficos imersivos diretamente no navegador, tendo em vista que a aplicação oferece uma experiência interativa e envolvente, adequada para dispositivos de XR como o Meta Quest. Apesar de desafios relacionados ao desempenho gráfico, otimizações específicas foram aplicadas para melhorar a fluidez e usabilidade e a integração de tecnologias web, como A-Frame e WebXR, mostrou-se eficaz para viabilizar a execução dessas experiências, tornando a aplicação mais acessível e prática para diversos contextos.

Este trabalho contribui para o campo emergente da visualização de dados imersiva, abordando tanto os desafios técnicos quanto as possibilidades interativas proporcionadas pela XR. Além de facilitar a análise de dados complexos, a aplicação desenvolvida serve como base para futuras pesquisas e desenvolvimentos que busquem expandir as capacidades de visualização e interação com dados em ambientes imersivos.

Palavras-chave: Realidade Estendida; Visualização de Dados; A-frame.

Abstract

The advancement of extended reality (XR) technologies, which encompass virtual reality (VR) and augmented reality (AR), has expanded the possibilities for data visualization, enabling immersive and interactive experiences that make the analysis of complex information more intuitive and accessible. This work presents the development of an immersive web application using the A-Frame framework, designed for XR-compatible devices. The project included the implementation of interactive graphs, such as bar charts, line graphs, scatter plots, donuts, and pie charts, allowing for the representation of complex data across multiple dimensions with a high degree of interactivity.

The methodology adopted in the project was based on the use of the Entity-Component-System (ECS) architecture to modularize and organize the code, in addition to employing data normalization methods to ensure uniformity in visualization. Custom components and integrated systems were developed to facilitate the creation of graphs and user interaction with the data, such as moving, resizing, rotating, and filtering information directly within the immersive environment. These elements were designed to be efficiently rendered in the browser, enhancing accessibility and eliminating the need for dedicated software or complex configurations.

The results demonstrated the feasibility of creating immersive graphs directly in the browser, as the application provides an interactive and engaging experience suitable for XR devices like Meta Quest. Despite challenges related to graphical performance, specific optimizations were applied to improve fluidity and usability, and the integration of web technologies, such as A-Frame and WebXR, proved effective in enabling the execution of these experiences, making the application more accessible and practical for various contexts.

This work contributes to the emerging field of immersive data visualization, addressing both the technical challenges and the interactive possibilities provided by XR. In addition to facilitating the analysis of complex data, the developed application serves as a foundation for future research and development aimed at expanding the capabilities for data visualization and interaction within immersive environments.

Keywords: Extended Reality; Data Visualization; A-frame.

Lista de Figuras

2.1	Espectro da realidade estendida	6
3.1	Logo do framework A-Frame	10
3.2	Logo do framework ThreeJS	11
3.3	Logo da linguagem de programação JavaScript	11
3.4	Logo da linguagem de marcação HTML	11
3.5	Logo do Blender	12
3.6	Logo do VSCode	12
3.7	Logo do Simple Web Server	12
3.8	Logo do Git	13
3.9	Logo do GitHub	13
3.10	Logo do Meta Quest	13
3.11	Controles do Meta Quest 2	14
3.12	Arquitetura do sistema	16
3.13	Menu principal	20
3.14	Menu principal com submenu dos gráficos ativo	21
3.15	Menu principal com submenu de interações ativo	21
3.16	Menu principal com estado de hover no submenu ativo	22
3.17	HUD explicativa para mover	22
3.18	HUD explicativa para rotacionar	22
3.19	HUD explicativa para redimensionar	23
3.20	HUD explicativa para filtrar	23
3.21	Bounding Box de cubo	24
3.22	Bounding Box de cilindro	24
3.23	Diagrama de hierarquia entre as entidades	24
3.24	Gráfico de barras com 3 dimensões	26
3.25	Gráfico de barras com 4 dimensões	26
3.26	Gráfico de dispersão com 3 dimensões	27
3.27	Gráfico de barras com 4 dimensões	27
3.28	Gráficos de pizza	27
3.29	Gráfico de donuts	28

3.30	Gráfico de linhas usando A-frame	29
3.31	Gráfico de linhas usando ThreeJS, versão inicial	30
3.32	Fluxograma do processo de mover um objeto	33
3.33	Fluxograma do processo de rotacionar um objeto	34
3.34	Fluxograma do processo de redimensionar um objeto	35
3.35	Fluxograma do processo de filtrar dados	36
3.36	Cenário principal	37
3.37	Cenário secundário	38
4.1	Menu de interações	43
4.2	Gráfico de barras com 3 dimensões	44
4.3	Gráfico de barras com 4 dimensões	44
4.4	Gráfico de dispersão com 3 dimensões	45
4.5	Gráfico de dispersão com 4 dimensões	45
4.6	Gráfico de pizza	46
4.7	Gráfico de donuts	46
4.8	Gráfico de linhas	47
4.9	Eixos com x e y	47
4.10	Eixos com x, y e z	48
4.11	Legenda de um gráfico	48
4.12	Interação de mover ativa.	49
4.13	Interação de rotacionar ativa.	49
4.14	Interação de redimensionar aumentando.	50
4.15	Interação de redimensionar diminuindo.	50
4.16	Interação de filtro ativa.	51
4.17	Antes e depois dos gráficos com o efeito de fake neon	51
4.18	Screenshots de instantes da animação	52
4.19	Modo AR no Meta Quest 3	52
4.20	Aplicação no Meta Quest 2 no modo VR.	53
4.21	Aplicação no Meta Quest 2 no modo AR.	53
4.22	Aplicação no Meta Quest 3 no modo VR.	54
4.23	Aplicação no Meta Quest 3 no modo AR.	54
4.24	Tela inicial da aplicação em um smartphone.	54
4.25	Tela inicial da aplicação em um notebook.	54

Lista de Tabelas

4.1	Comparação dos dados do modelo 3D antes e depois da otimização	41
4.2	Resultados dos testes em diferentes dispositivos.	54

Lista de Abreviaturas

VR	<i>Virtual Reality</i>
AR	<i>Augmented Reality</i>
XR	<i>Extended Reality</i>
MR	<i>Mixed Reality</i>
AV	<i>Augmented Virtuality</i>
WebXR	<i>Realidade Estendida na Web</i>
WebVR	<i>Realidade Virtual na Web</i>
JS	<i>JavaScript</i>
CSV	<i>Comma-Separated Values</i>
3D	<i>Three Dimensional</i>
HTML	<i>HyperText Markup Language</i>
ECS	<i>Entity Component System</i>
HUD	<i>Heads-Up Display</i>
UX	<i>User Experience</i>
HSL	<i>Hue, Saturation, Light</i>
API	<i>Application Programming Interface</i>
FPS	<i>Frame Per Second</i>

Sumário

1	Introdução	1
1.1	Motivação	3
1.2	Objetivos	3
1.2.1	Objetivo Geral	3
1.2.2	Objetivos Específicos	3
1.3	Organização do Trabalho	4
2	Fundamentação Teórica	5
2.1	Realidade Estendida	5
2.2	Visualização de dados	6
2.3	Análise de dados imersiva	6
2.4	Interações em tecnologias imersivas	7
2.5	Aplicações imersivas na web	8
3	Metodologia	10
3.1	Tecnologias e ferramentas utilizadas	10
3.2	Estrutura do código e arquitetura do sistema	14
3.2.1	Estrutura do Código	14
3.2.2	Camadas do Sistema	15
3.2.3	Manutenção e Escalabilidade	15
3.2.4	Arquitetura do Sistema	16
3.3	Integração de realidade virtual e aumentada	16
3.3.1	Configuração e inicialização da cena	17
3.3.2	Execução da aplicação	17
3.4	Preparação e tratamento dos dados	18
3.4.1	Leitura dos dados	18
3.4.2	Normalização	18
3.5	Interação com os usuários e feedback	19
3.5.1	Menu principal	19
3.5.2	HUD	22
3.5.3	Bounding Box	23

3.6	Gráficos Implementados	25
3.6.1	Gráfico de Barras	26
3.6.2	Gráfico de Dispersão	26
3.6.3	Gráfico de Pizza	27
3.6.4	Gráfico de Donuts	28
3.6.5	Gráfico de Linhas	28
3.6.6	Componentes auxiliares	31
3.7	Interações com os gráficos	32
3.7.1	Mover	32
3.7.2	Rotacionar	33
3.7.3	Redimensionar	34
3.7.4	Filtrar	35
3.8	Aparência	36
3.8.1	Modelo 3D e otimização	36
3.8.2	Ambiente secundário	37
3.8.3	Efeito neon	38
3.8.4	Animações	38
4	Resultados	40
4.1	Dificuldades enfrentadas	40
4.1.1	FPS e Otimização	40
4.1.2	Problemas com a lógica, eventos e estados	41
4.1.3	Lógica matemática para gerar os gráficos	42
4.1.4	Textos ocultos	42
4.2	Aplicação Desenvolvida	42
4.2.1	Funcionalidades	42
4.2.2	Demonstrações em vídeo	53
4.3	Testes em diferentes dispositivos	53
5	Conclusão	55
	Bibliografia	58

Capítulo 1

Introdução

Hoje em dia, mais dados são gerados e armazenados do que em toda a história da humanidade, crescendo em uma velocidade sem precedentes. Embora esse volume de informações traga oportunidades, também apresenta desafios complexos, como o armazenamento, gerenciamento e interpretação desses dados para extrair conhecimento útil Chen (2005), Olshannikova et al. (2015). Nesse contexto, a visualização de dados se destaca como uma ferramenta essencial, a fim de transformar dados brutos em percepções relevantes e acessíveis.

Os autores Shakeel et al. (2022) e Donalek et al. (2014) ressaltam que o ser humano tem maior facilidade em aprender, observar padrões e reter informações quando essas são apresentadas visualmente. Tabelas extensas ou arquivos de texto, embora possam conter dados valiosos, tem o seu potencial reduzido caso não permitam identificar padrões de maneira intuitiva Donalek et al. (2014). Além disso, a falta de compreensão dos dados pode levar a interpretações equivocadas e decisões errôneas. Dessa forma, de acordo com Donalek et al. (2014), uma visualização de dados eficiente é a ponte entre o conteúdo quantitativo dos dados e a intuição humana e, portanto, um componente essencial no caminho científico de transformar um dado em conhecimento e entendimento.

No entanto, a visualização de dados deve ser usada em conjunto com outras técnicas de processamento de dados, a fim de facilitar a interpretação e levar a percepções que, de outra forma, seriam mais difíceis de alcançar. Diversas técnicas de visualização já são amplamente utilizadas para transformar dados complexos em representações visuais claras e intuitivas, sendo muitas delas baseadas em computação gráfica Chen (2005). Estudos como Chen (2005) defendem que a técnica a ser usada depende do tipo e da dimensão dos dados a serem estudados, como, por exemplo, visualização de volume para dados volumétricos ou visualização de escoamentos para dinâmica de fluidos computacional.

Essa versatilidade na aplicação das técnicas de visualização também se reflete no mercado, com empresas investindo cada vez mais na área e resultando em ferramentas amplamente adotadas Olshannikova et al. (2015). Exemplos disso incluem aplicações como Google Charts, D3.js e Plotly, que ajudam usuários e empresas a explorar dados de ma-

neira eficaz. Nesse contexto, a realidade estendida (XR, do inglês *Extended Reality*) surge como uma tecnologia promissora para ampliar as capacidades tradicionais, permitindo formas inovadoras de representar e interagir com informações Tamayo et al. (2018).

Algumas aplicações já foram desenvolvidas explorando essas tecnologias. Por exemplo, BabiaXR¹ é uma ferramenta que utiliza a realidade virtual para facilitar a exploração e análise de dados tridimensionais Moreno-Lumbreras et al. (2023, 2022). A-Frame Charts Component² é um componente baseado em A-Frame que facilita a criação de gráficos tridimensionais para WebXR. Ferramentas comerciais, como Virtualitics³, oferecem ambientes colaborativos em realidade virtual para análise avançada de dados Hadjar et al. (2018), enquanto Flow Immersive⁴ foca em visualizações interativas para *storytelling* de dados em realidade aumentada DiBenigno et al. (2021). Por fim, LookVR⁵ permite criar experiências interativas em realidade virtual para diversos tipos de visualização Dibbern et al. (2018).

Assim, soluções que utilizam tecnologias XR ampliam as possibilidades de visualização ao criar ambientes tridimensionais imersivos, onde os usuários podem interagir diretamente com os dados. Isso permite identificar padrões, tendências e anomalias com maior facilidade Moran et al. (2015). Com isso, um grande benefício do uso de técnicas de análise de dados imersiva em comparação com técnicas de visualização tradicionais está na expansão do mundo 3D.

O autor Chen (2005) lista as principais vantagens e o papel desempenhado por cada uma delas, como imersão, interação multissensorial, navegação, sensação de presença e exibições multimodais. Por meio dessas características particulares, o uso de XR na visualização de dados pode representar uma evolução na área e proporcionar uma experiência mais rica e intuitiva. Quando essas características são exploradas em conjunto, o verdadeiro potencial do uso da realidade estendida se destaca, padrões são mais facilmente identificados e os usuários se tornam mais capacitados para lidar com dados cada vez mais complexos.

Além das vantagens intrínsecas da realidade estendida, sua integração na web amplia ainda mais seu potencial ao proporcionar maior acessibilidade. Isso permite que usuários explorem visualizações imersivas de forma prática, sem necessidade de instalações complexas e com conectividade direta a fontes de dados externas. Embora, conforme apontado por Liu et al. (2023), aplicações web possam enfrentar limitações de desempenho em relação a soluções *standalone*, a compatibilidade entre plataformas, a facilidade de atualização e a possibilidade de colaboração remota tornam essa abordagem especialmente valiosa em cenários que envolvem análises dinâmicas e interativas em tempo real.

¹<https://babiaxr.gitlab.io/>

²<https://adrixp.github.io/>

³<https://www.virtualitics.com>

⁴<https://www.flowimmersive.com>

⁵<https://store.steampowered.com/app/595490/LookVR/>

1.1 Motivação

A visualização de dados tem se mostrado uma ferramenta indispensável para a interpretação e compreensão de grandes volumes de informações. Tecnologias imersivas oferecem alternativas inovadoras para ampliar as possibilidades dessa área, permitindo interações, análises aprofundadas e experiências mais engajantes. Dessa forma, o uso de realidade estendida na visualização de dados representa um campo em crescimento, com grande potencial de transformação e resultados promissores. Essas tecnologias possibilitam um engajamento multissensorial e exploram perspectivas multidimensionais que superam as limitações das abordagens bidimensionais tradicionais.

Apesar do progresso, ainda há desafios consideráveis, como a falta de padrões bem definidos, limitações na interação com dados complexos e uma adoção restrita a nichos específicos Wahidi et al. (2023). Este trabalho busca preencher parte dessa lacuna ao explorar o desenvolvimento de uma aplicação baseada em XR para dados, aumentando a acessibilidade e permitindo que usuários interajam de forma mais intuitiva com informações complexas.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é estudar o uso da realidade estendida em conjunto com a área de visualização de dados e desenvolver uma aplicação web que permita a análise interativa e imersiva de dados em diferentes dispositivos, com foco especial em dispositivos de realidade virtual e aumentada. O trabalho busca contribuir para este campo emergente, seja por meio das ferramentas utilizadas, das técnicas implementadas, das interações desenvolvidas ou do projeto como um todo.

1.2.2 Objetivos Específicos

Para atingir o objetivo geral, foram estabelecidos os seguintes objetivos específicos:

- Desenvolver uma aplicação imersiva em realidade estendida utilizando o framework A-frame;
- Garantir que a aplicação seja compatível com dispositivos que possuam navegadores web, especialmente os de realidade virtual e aumentada;
- Implementar diferentes tipos de gráficos (barras, linhas, dispersão, etc.) para permitir a visualização de dados;

- Oferecer interações intuitivas com os gráficos, como mover, redimensionar, filtrar e rotacionar;
- Implementar uma abordagem que dispense bibliotecas externas para leitura de dados e renderização de gráficos, com foco no controle total sobre o código;
- Assegurar que a aplicação consiga interpretar dados no formato CSV, com feedback apropriado em caso de erro;
- Adotar a metodologia de desenvolvimento ECS (Entity-Component-System) para maior modularidade e organização do código;
- Desenvolver um código que facilite a instanciação de novos gráficos e abstraia a complexidade do restante do código.

1.3 Organização do Trabalho

Este trabalho está estruturado em quatro seções principais, além desta introdução. A Seção 2, intitulada fundamentação teórica, apresenta conceitos e bases teóricas relevantes para o desenvolvimento da aplicação proposta, incluindo realidade estendida, visualização de dados, experiência do usuário, análise de dados imersiva e aplicações imersivas na web.

A Seção 3, referente à metodologia, descreve detalhadamente as abordagens e ferramentas utilizadas no projeto. Inicialmente, são apresentados os recursos tecnológicos empregados, seguidos pela explicação da estrutura do código e arquitetura do sistema. A integração de realidade virtual e aumentada é explorada, abordando aspectos como configuração de cena, comportamentos específicos e execução em diferentes contextos. Também são descritos os processos de preparação e tratamento dos dados, interação com usuários e feedback, além de especificações dos gráficos implementados e suas interações.

Na Seção 4, são descritos os resultados obtidos durante o desenvolvimento do projeto. São relatadas as dificuldades enfrentadas durante o desenvolvimento, como questões de otimização e lógica matemática, além de problemas específicos com textos e eventos. Em seguida, são apresentadas as funcionalidades e características da aplicação desenvolvida, avaliando seu desempenho e usabilidade.

Por fim, as conclusões são apresentadas na Seção 5, onde são discutidos os resultados obtidos, as contribuições do trabalho e perspectivas futuras para aprimoramento e continuidade do projeto.

Capítulo 2

Fundamentação Teórica

Nesta seção, serão descritas as principais áreas relacionadas, ferramentas e tecnologias utilizadas neste trabalho.

2.1 Realidade Estendida

A realidade estendida é um termo guarda-chuva que abrange tecnologias imersivas, como realidade virtual, realidade aumentada, realidade mista e virtualidade aumentada Liu et al. (2022). O conceito foi inicialmente introduzido como uma forma de descrever os diferentes níveis de interações entre o mundo real e o virtual, ampliando a percepção e a interação dos usuários com seus ambientes.

No espectro da realidade estendida, a realidade virtual (VR, do inglês *Virtual Reality*) substitui totalmente o ambiente real por um ambiente virtual, geralmente composto por cenários tridimensionais ou vídeos 360°. Já a realidade aumentada (AR, do inglês *Augmented Reality*) insere elementos virtuais no mundo real, proporcionando informações adicionais ou objetos interativos em tempo real. A virtualidade aumentada (AV do inglês *Augmented Virtuality*), por sua vez, consiste em ambientes virtuais que incorporam elementos do mundo real, como imagens ou dados capturados em tempo real, criando uma fusão onde o real é complementado pelo virtual. Por fim, a realidade mista (MR, do inglês *Mixed Reality*) combina ambos os mundos, permitindo interações dinâmicas entre objetos virtuais e reais.

A imagem 2.1 ilustra os conceitos explicados acima, evidenciando como cada tecnologia se posiciona em relação ao equilíbrio entre elementos reais e virtuais. Essa representação facilita a compreensão e destaca as diferenças e intersecções entre as subáreas.

Essas tecnologias têm sido amplamente aplicadas em diversos setores, como educação, saúde, entretenimento e engenharia, trazendo benefícios para superar obstáculos como tempo, espaço, segurança e recursos operacionais Vasarainen et al. (2021). Este trabalho concentra-se no desenvolvimento de uma aplicação em realidade virtual e aumentada, explorando as capacidades únicas de cada uma.

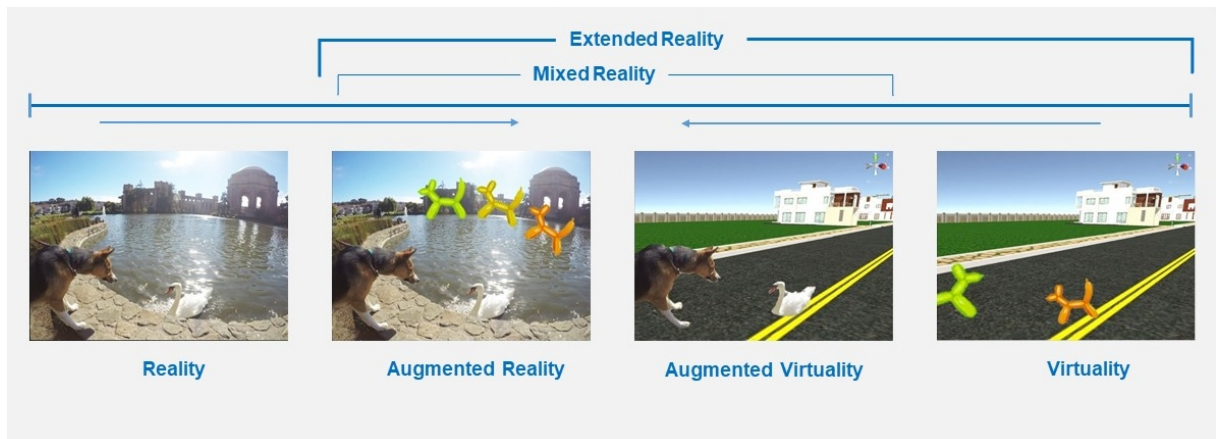


Figura 2.1: Espectro da realidade estendida
Fonte: XR for all

2.2 Visualização de dados

A visualização de dados, também conhecida como *data visualization*, é a prática de transformar informações abstratas e muitas vezes complexas em representações visuais claras e compreensíveis. Historicamente, esse campo começou com gráficos básicos, como tabelas e diagramas, evoluindo para formas mais complexas com o avanço das tecnologias computacionais. Ferramentas modernas permitem representar grandes volumes de dados com alta eficiência e impacto visual.

Essa área desempenha um papel crucial ao auxiliar na análise e comunicação de informações, destacando padrões, tendências e *outliers* que poderiam passar despercebidos. Métodos como gráficos de dispersão, linhas e pizza são amplamente usados para facilitar a tomada de decisões informadas em áreas como negócios, saúde e ciência. Portanto, é uma ferramenta fundamental em diversas áreas do conhecimento.

Além disso, a visualização de dados contribui para democratizar o acesso à informação, tornando dados acessíveis até mesmo para públicos não técnicos. A simplicidade visual e a clareza ajudam a transmitir mensagens complexas de maneira eficaz, promovendo maior engajamento e compreensão Shakeel et al. (2022).

De acordo com Moran et al. (2015), a representação visual de dados promove *insights* mais rápidos e interação mais significativa com as informações. Isso é particularmente relevante quando se trata de identificar anomalias ou padrões ocultos que podem não ser evidentes em formatos textuais, tabulares ou em gráficos tradicionais.

2.3 Análise de dados imersiva

A análise de dados imersiva, também conhecida como *Immersion Analytics*, é uma área emergente que busca explorar as vantagens das tecnologias de realidade estendida

para representar e analisar dados. Esse campo se concentra em reduzir a barreira entre pessoas, dados e ferramentas, criando experiências interativas que permitem decisões mais informadas e *insights* mais profundos.

A maioria das técnicas de visualização de dados foi desenvolvida para telas de computadores. No entanto, o avanço acelerado das tecnologias abriu novas oportunidades para explorar, compreender e apresentar dados de maneiras inovadoras Lee et al. (2019). As visualizações imersivas, habilitadas por tecnologias de realidade aumentada e realidade virtual, têm se mostrado promissoras para superar as limitações das abordagens tradicionais, permitindo visualização científica em ambientes tridimensionais, narrativas interativas para apresentação de informações, escritórios virtuais para colaboração e exploração de dados.

Ao contrário das abordagens tradicionais de análise de dados, que muitas vezes dependem de gráficos estáticos ou painéis 2D, a análise imersiva utiliza ambientes virtuais e aumentados para proporcionar uma experiência tridimensional. Com isso, ela amplia a capacidade cognitiva dos usuários, promovendo uma compreensão mais intuitiva de padrões e correlações.

No entanto, embora promissora, a visualização imersiva apresenta desafios, como a necessidade de projetar interações naturais e intuitivas e a representação eficaz de dados complexos. Assim, a melhoria contínua nas técnicas de visualização e interação é essencial para enfrentar essas barreiras.

2.4 Interações em tecnologias imersivas

A experiência do usuário (UX, do inglês *User Experience*) em ambientes imersivos, representa um campo crescente de estudo e aplicação. Ao contrário das interfaces tradicionais, onde a interação é feita por meio de mouse, teclado ou toque, a interação em ambientes imersivos é altamente dinâmica, permitindo ao usuário explorar de uma maneira mais intuitiva e envolvente.

No contexto da visualização e análise de dados imersiva, os usuários têm a oportunidade de interagir diretamente com os dados de formas novas e mais sensoriais. Isso significa que, ao invés de apenas observar gráficos ou tabelas em uma tela plana, o usuário pode manipular objetos virtuais, navegar em cenas tridimensionais ou até mesmo se mover fisicamente no espaço, explorando diferentes perspectivas e níveis de detalhamento dos dados. A realidade aumentada e a realidade virtual oferecem, assim, uma gama de novas possibilidades para a interação, tornando o processo de análise mais envolvente, interativo e eficaz.

Dessa forma, a criação de experiências imersivas de qualidade não se limita à visualização, mas também envolve a maneira como o usuário pode interagir com esses dados. Com o aumento da complexidade e da quantidade de informações, o desafio se

torna proporcionar maneiras de explorar e analisar dados em grande escala de forma acessível e compreensível. A interação em AR e VR pode facilitar essa tarefa, permitindo ao usuário manipular grandes volumes de dados de maneira visual, eficiente e, muitas vezes, mais intuitiva do que as interfaces tradicionais.

Sendo assim, para que uma experiência imersiva seja bem-sucedida, é essencial que as interações sejam fluídas, naturais e responsivas, respeitando a forma como o usuário está acostumado a interagir com o ambiente físico. Hoje, é possível criar experiências em que o usuário pode não apenas visualizar e manipular dados, mas também interagir com eles por meio de ações físicas naturais, como apontar, pegar, empurrar ou até mesmo aplicar gestos para acionar comandos.

Com a evolução das tecnologias de rastreamento de movimento, detecção de gestos e reconhecimento de voz, as possibilidades de interação tem se ampliado ainda mais. Isso pode incluir o uso de controladores de movimento, gestos ou mesmo rastreamento ocular, permitindo que o usuário interaja com os dados de maneira totalmente intuitiva. Tais interações tornam o processo de exploração de dados mais envolvente e, muitas vezes, mais produtivo.

Além disso, em ambientes imersivos, o *feedback* fornecido ao usuário — seja visual, auditivo ou tátil — tem um papel fundamental na melhoria da experiência de interação. Um *feedback* eficiente e imediato ajuda a confirmar a ação realizada pelo usuário, reduzindo a incerteza e aumentando a confiança ao realizar tarefas complexas, como análise e manipulação de dados em tempo real.

Dessa forma, essas interações criam um ambiente onde os usuários podem explorar dados de maneira mais profunda e com maior controle, transformando a maneira como as decisões são tomadas e como as informações são compreendidas.

2.5 Aplicações imersivas na web

O avanço das tecnologias de realidade virtual e aumentada trouxe mudanças relevantes para a criação de experiências imersivas, com destaque para a transição dessas experiências para a web. A criação de aplicações imersivas tradicionais, utilizando plataformas como Unity, por exemplo, exigia a instalação de software específico e a utilização de dispositivos dedicados, como óculos VR ou AR. No entanto, a web tem se tornado um ambiente cada vez mais propício para esse tipo de experiência, oferecendo uma série de vantagens, como a acessibilidade direta através de navegadores, sem a necessidade de instalações ou equipamentos caros.

Uma das principais tecnologias que possibilita a criação de experiências imersivas diretamente na web é a API WebXR. O WebXR permite que navegadores ofereçam suporte tanto para VR quanto para AR, tornando possível a interação com ambientes 3D imersivos em dispositivos móveis, *desktops* e *headsets* de realidade virtual, tudo sem a necessidade

de *softwares* adicionais. Essa abordagem abre a porta para uma audiência muito maior, já que as pessoas podem acessar experiências imersivas diretamente do navegador, utilizando dispositivos comuns, como *smartphones* ou PCs.

Outra tecnologia que tem se destacado no desenvolvimento de experiências de realidade virtual na web é o A-Frame. Baseado na biblioteca Three.js, ele oferece um framework simples para a criação de cenas 3D interativas usando apenas HTML e JavaScript. Por sua vez, o Three.js fornece as ferramentas para renderizar gráficos 3D avançados em navegadores, complementando o A-Frame ao possibilitar personalizações gráficas mais profundas e efeitos visuais sofisticados.

Embora a utilização de Unity e outras ferramentas de desenvolvimento de ambientes virtuais ainda seja a escolha preferencial para muitas aplicações de VR/AR devido ao seu poder gráfico, as tecnologias para a web têm mostrado grandes avanços. Com isso, mesmo que haja uma limitação das aplicações imersivas para a web relacionada ao desempenho gráfico e a renderização de gráficos 3D avançados, elas ainda representam uma alternativa competitiva e acessível aos métodos tradicionais.

Capítulo 3

Metodologia

A metodologia adotada neste trabalho envolve a utilização de diversas tecnologias e técnicas para o desenvolvimento de uma aplicação interativa de visualização de dados em realidade virtual e aumentada. A seção a seguir descreve detalhadamente os procedimentos e recursos empregados para a construção da aplicação, incluindo a escolha das tecnologias, a organização do código, as camadas do sistema e as estratégias de integração de VR/AR. Além disso, são apresentados os gráficos implementados, as interações desenvolvidas e alguns componentes essenciais da interface, visando proporcionar uma experiência imersiva e dinâmica ao usuário.

3.1 Tecnologias e ferramentas utilizadas

Esta seção descreve as tecnologias e ferramentas empregadas no desenvolvimento da aplicação, detalhando suas finalidades e contribuições para o projeto.

- A-Frame: É um framework de código aberto baseado no ThreeJS, projetado para facilitar o desenvolvimento de experiências de realidade virtual e aumentada. Ele abstrai a complexidade da criação de cenas 3D, permitindo que os desenvolvedores criem aplicações imersivas de forma simples e eficiente. No projeto, o A-Frame foi utilizado como a espinha dorsal para construir a interface e permitir a integração com os dispositivos. Na Figura 3.1, é exibido o logo do *framework* A-Frame.

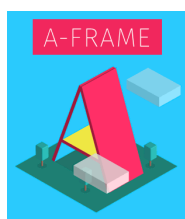


Figura 3.1: Logo do framework A-Frame
Fonte: Wikimedia Commons

- Three.js: É uma biblioteca JavaScript que oferece uma poderosa API para renderização de gráficos 3D em navegadores. Com seu vasto conjunto de funcionalidades, como controle de iluminação, animação e geometria 3D, o Three.js complementa o A-Frame, permitindo a adição de efeitos visuais avançados e personalizações gráficas. No projeto, foi usado para a criação de geometrias mais complexas, de maneira simplificada e otimizada. O logo do Three.js pode ser visto na Figura 3.2.

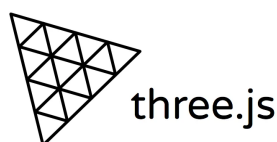


Figura 3.2: Logo do framework ThreeJS
Fonte: Xtreme Thoughts

- JavaScript: É a principal linguagem de programação utilizada para implementar a lógica de interação da aplicação. Sua versatilidade permite o controle de eventos, manipulação de objetos e integração entre diferentes bibliotecas e *frameworks*. No projeto, JavaScript foi essencial para orquestrar a interação entre os elementos criados no A-Frame e Three.js, gerenciar os estados da aplicação e garantir uma experiência de usuário fluida e dinâmica. A Figura 3.3 apresenta o seu logo.



Figura 3.3: Logo da linguagem de programação JavaScript
Fonte: PNGWing

- HTML: É uma linguagem de marcação fundamental para a construção de páginas web. Ele proporciona a estrutura básica para a aplicação, organizando os elementos visuais e a interface do usuário. No projeto, o HTML foi utilizado para integrar o A-Frame e Three.js em uma página web, e proporcionar a interface da experiência de VR/AR. A Figura 3.4 ilustra o logo do HTML.



Figura 3.4: Logo da linguagem de marcação HTML
Fonte: Wikipedia

- Blender: É um *software* de modelagem e animação 3D altamente versátil, utilizado para criar, texturizar e animar modelos tridimensionais. Sua importância no projeto foi fundamental para otimizar os elementos 3D que enriqueceram a cena e aumentaram a imersão, melhorando o desempenho dos mesmos no ambiente virtual. Na Figura 3.5, é exibido o logo do Blender.



Figura 3.5: Logo do Blender
Fonte: Wikipedia

- Visual Studio Code (VSCode): É um editor de código-fonte popular e altamente poderoso, com suporte para diversas extensões. Ele foi utilizado para escrever o código do projeto. Com sua integração com Git, VSCode facilitou o controle de versão, além de oferecer recursos que melhoraram a produtividade durante o desenvolvimento. A Figura 3.6 exibe o logo do VSCode.



Figura 3.6: Logo do VSCode
Fonte: PNGEgg

- Simple Web Server: É uma ferramenta utilizada para hospedar a aplicação durante o desenvolvimento. Este servidor facilitou o ciclo de desenvolvimento, ao possibilitar a visualização contínua das cenas e a realização de ajustes em tempo real, sem a necessidade de configurações complexas. Além disso, possibilitou a visualização em tempo real nos dispositivos de XR, permitindo que as alterações no código fossem testadas rapidamente. O logo do Simple Web Server está ilustrado na Figura 3.7.



Figura 3.7: Logo do Simple Web Server
Fonte: Wikipedia

- Git: É um sistema de controle de versão distribuído, essencial para acompanhar e gerenciar as alterações no código, mantendo um histórico completo do desenvolvimento. Sua utilização no projeto garantiu organização e rastreamento das modificações realizadas durante o desenvolvimento da aplicação. O logo do Git está ilustrado na Figura 3.8.



Figura 3.8: Logo do Git

Fonte: DIO

- GitHub: É uma plataforma online que facilita o armazenamento e compartilhamento de código. No projeto, o GitHub foi utilizado para hospedar o código e o histórico gerado pelo Git, garantindo acessibilidade e colaboração entre os membros da equipe. A Figura 3.9 ilustra o logo do GitHub.



Figura 3.9: Logo do GitHub

Fonte: Medium

- Meta Quest 2: Óculos de realidade virtual e aumentada altamente imersivo, usados para testar e validar a experiência de VR/AR. Esse dispositivo foi crucial para avaliar a interação, usabilidade e qualidade visual da aplicação em um ambiente real. O uso do Meta Quest garantiu que as funcionalidades estivessem otimizadas para a experiência final, ajudando a ajustar a qualidade visual e a fluidez das interações dentro do ambiente XR. Na Figura 3.10, é exibido o logo do Meta Quest 2.



Figura 3.10: Logo do Meta Quest

Fonte: MIXED Reality News

Já na Figura 3.11, estão destacados os botões dos controles do dispositivo, com seus respectivos nomes e funcionalidades, que serão utilizados durante a interação com a aplicação.

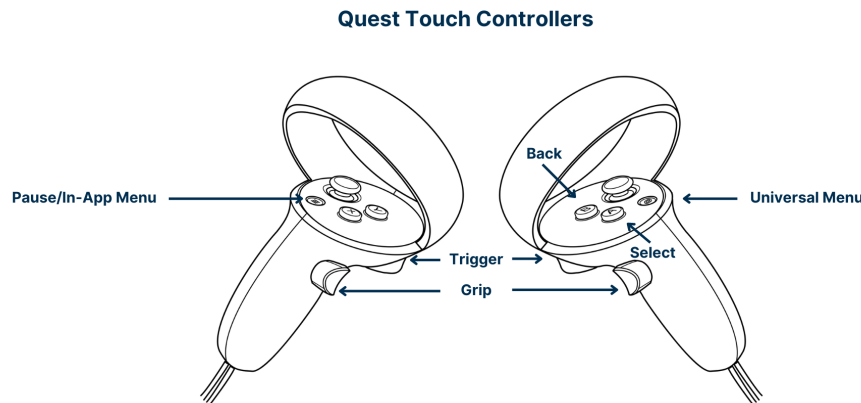


Figura 3.11: Controles do Meta Quest 2
Fonte: Meta Quest 2 User Guide

3.2 Estrutura do código e arquitetura do sistema

O sistema foi projetado para oferecer flexibilidade, escalabilidade e facilidade de manutenção. Esta seção descreve a estrutura do código e a arquitetura do sistema, destacando como os diferentes componentes interagem entre si para fornecer uma experiência de usuário otimizada.

3.2.1 Estrutura do Código

O código do projeto foi estruturado de forma a facilitar a manutenção, legibilidade e escalabilidade, seguindo princípios de boas práticas em desenvolvimento de *software*. A organização geral do código adota uma estrutura modular, separando funcionalidades em diretórios específicos para melhorar o isolamento e o reuso de componentes. Abaixo, são apresentados os principais diretórios e seus conteúdos:

- **assets/** – Contém arquivos de recursos essenciais, como imagens, ícones, modelos 3D, fontes, arquivos de dados e outros elementos visuais. A separação entre esses recursos e o código facilita a organização do projeto, mantendo uma estrutura mais limpa e modular.
- **components/** – Este diretório armazena componentes reutilizáveis, especialmente aqueles que representam partes da interface de usuário no ambiente VR/AR. Cada

componente possui um arquivo próprio, o que garante flexibilidade na integração e permite customizações independentes. Exemplos incluem os componentes de gráficos, e os auxiliares, como eixos e legendas.

- **scripts/** – Abriga arquivos JavaScript responsáveis pela implementação da lógica da aplicação. Esses arquivos gerenciam eventos, animações e interações específicas na experiência VR/AR, controlando o comportamento da interface e dos elementos interativos.
- **systems/** – Contém os arquivos referentes aos sistemas responsáveis pela gestão e controle dos gráficos e dados.

3.2.2 Camadas do Sistema

A arquitetura do sistema foi projetada para suportar uma aplicação VR/AR interativa, onde cada módulo possui responsabilidades bem definidas e se comunica com outros de maneira eficiente. A arquitetura é composta por uma camada de interface do usuário e uma camada de lógica de aplicação, conforme descrito a seguir:

- **Camada de interface do usuário** — Implementada principalmente com A-Frame e Three.js, essa camada gerencia a apresentação dos elementos em 3D. O A-Frame foi utilizado como *framework* principal, enquanto o Three.js complementa a renderização.
- **Camada de lógica de aplicação** — Desenvolvida em JavaScript, essa camada orquestra a interação dos usuários com os elementos virtuais. Ela também gerencia o estado da aplicação e controla as ações desencadeadas por eventos, como a seleção e manipulação de objetos na cena VR/AR.

3.2.3 Manutenção e Escalabilidade

O sistema foi estruturado visando a escalabilidade e facilidade de manutenção. A divisão em camadas e módulos bem definidos facilita a depuração e atualização do código. Além disso, foi adotado o modelo Entity-Component-System (ECS) para a organização do código e gerenciamento de objetos na aplicação. Esse padrão arquitetural favorece a desacoplagem e a flexibilidade, permitindo uma melhor performance ao lidar com grandes quantidades de dados e interações complexas.

O ECS proporciona um alto nível de modularização, facilitando a adição de novos recursos, além de melhorar a manutenção do código, pois permite que funcionalidades e comportamentos sejam separados e atualizados de forma independente. Com essa abordagem, o sistema tem a capacidade de expandir e modificar partes da aplicação sem comprometer o funcionamento do todo.

Essa estrutura, aliada à organização modular do código, fornece uma base para futuras expansões e otimizações, garantindo uma experiência de usuário consistente.

3.2.4 Arquitetura do Sistema

A arquitetura do sistema integra várias tecnologias que trabalham juntas para garantir a imersão e a interatividade da aplicação. A camada cliente, composta pela página web, utiliza HTML, JavaScript, A-Frame, Three.js e a API WebXR para renderizar a cena e permitir interações no ambiente VR/AR. A comunicação com os dispositivos XR é feita através da API WebXR, permitindo que o usuário interaja diretamente com a página web e os elementos da cena 3D. O diagrama 3.12 ilustra a interação entre o usuário, o dispositivo XR e as tecnologias que gerenciam a interface e a renderização da aplicação.

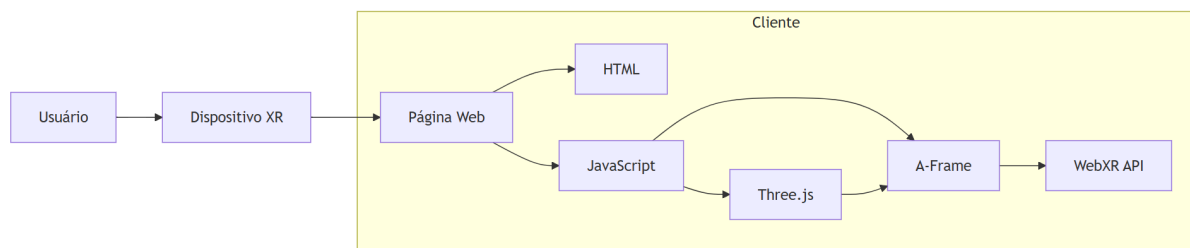


Figura 3.12: Arquitetura do sistema
Fonte: Autoria própria

3.3 Integração de realidade virtual e aumentada

A estrutura base da aplicação utiliza a biblioteca A-Frame, que fornece suporte nativo para VR e AR e permite a criação de ambientes tridimensionais interativos diretamente no navegador. Essa abordagem centraliza as interações e configurações da cena, permitindo que a aplicação seja facilmente adaptada para diferentes dispositivos e cenários de uso. Nesta seção, são detalhadas a configuração da cena, o processo de execução da aplicação e as diferenças no comportamento entre os modos VR e AR.

3.3.1 Configuração e inicialização da cena

Para configurar a cena, foi utilizado o elemento principal `<a-scene>`, que serve como o *container* para todos os objetos, luzes e elementos interativos. Para habilitar o modo de realidade estendida, foi adicionado o atributo `xr-mode-ui="enabled: true; XRMode: xr"`, do próprio A-frame, responsável por habilitar o modo XR, possibilitando o funcionamento tanto em VR quanto em AR.

Na tela inicial, botões interativos do próprio a-frame permitem que o usuário escolha entre os modos antes de entrar na cena, proporcionando uma interface intuitiva para navegação. A escolha do modo determina como a cena será carregada, bem como quais elementos e funcionalidades adicionais serão ativados.

As configurações específicas para VR e AR incluem diferenças importantes para otimizar a experiência do usuário. Por exemplo, o componente `hide-on-enter-vr` é utilizado para ocultar elementos como céu, chão e modelos 3D dos ambientes ao entrar no modo AR. Esse ajuste evita sobrecarga visual, garantindo uma interação mais natural com o ambiente físico.

Adicionalmente, foi desenvolvido um *script* auxiliar para otimizar a performance no modo AR. Esse *script* libera recursos dos elementos desnecessários automaticamente, melhorando o desempenho da aplicação.

3.3.2 Execução da aplicação

Para acessar a aplicação, o usuário pode utilizar um endereço local ou um servidor online onde a aplicação esteja hospedada. Ao iniciar, a tela inicial apresenta as opções de modo, possibilitando que o usuário escolha entre VR ou AR conforme o dispositivo em uso.

No modo VR, a cena é carregada de forma imersiva, criando um ambiente tridimensional completo ao redor do usuário. Este modo aproveita ao máximo os *headsets* VR, proporcionando uma experiência envolvente e interativa, com o usuário podendo explorar e interagir no espaço virtual.

No modo AR, a aplicação utiliza a câmera do dispositivo para sobrepor elementos virtuais ao ambiente real, proporcionando uma interação fluida com o espaço físico ao redor. O usuário pode explorar e interagir com a cena virtual integrada ao mundo real, criando uma experiência mais híbrida e prática para dispositivos móveis compatíveis.

Em dispositivos móveis, a aplicação oferece uma experiência tridimensional sem ativar a câmera. Neste caso, o usuário pode visualizar o ambiente 3D e movimentar-se pelo espaço, mas sem a possibilidade de interagir diretamente com os elementos virtuais, como ocorre nos modos VR e AR completos.

3.4 Preparação e tratamento dos dados

Para a implementação dos gráficos na aplicação, foi adotado o formato CSV (comma-separated values) para os dados. Esse tipo de arquivo é amplamente utilizado para o armazenamento de dados estruturados devido à sua simplicidade e versatilidade, sendo facilmente interpretado por diversas ferramentas de análise e visualização de dados. Além disso, a maioria dos sistemas e aplicações permite a exportação de dados para CSV, o que torna a aplicação mais acessível e adaptável a diferentes fontes de dados. Atualmente, a aplicação não permite a importação de arquivos de forma dinâmica, ao invés disso, os arquivos CSV são definidos na linha de código responsável pela criação dos gráficos e armazenados diretamente na pasta de *assets*, onde o *script* responsável pela leitura localiza, processa e envia para o componente do gráfico correspondente. Nesta seção, serão descritos os processos de leitura e tratamento dos dados.

3.4.1 Leitura dos dados

Na criação de um gráfico por linha de código, é especificado como atributo obrigatório o nome do arquivo csv cujos dados serão exibidos. Por padrão, são lidas as três primeiras colunas e associadas respectivamente aos eixos X, Y e Z, com a última coluna sendo automaticamente utilizada como legenda dos dados. É possível alterar a quantidade e a ordem das colunas, a fim de resultar em uma visualização personalizada.

Além disso, também é possível limitar o número de linhas exibidas simultaneamente no gráfico. Essa quantidade máxima de linhas é ajustável, permitindo que o usuário trabalhe tanto com amostras pequenas quanto com grandes volumes de dados, mantendo a interface fluida. Esse método permite maior controle sobre o que será exibido e facilita o ajuste dos dados conforme necessário, também melhorando o desempenho e evitando a sobrecarga cognitiva.

Os dados de referência utilizados ao longo desse trabalho incluem métricas relacionadas à quantidade de vendas por mês/ano e à relação de preços com características em residências. No entanto, a aplicação suporta qualquer tipo de CSV, contanto que possua colunas numéricas ou categóricas adequadas aos eixos de um gráfico.

3.4.2 Normalização

A normalização é um processo essencial para a padronização dos dados exibidos nos gráficos, garantindo que valores de diferentes escalas possam ser manipulados de forma proporcional. A técnica de normalização aplicada redimensiona os valores para o intervalo de 0 a 1, o que facilita a visualização e torna o gráfico mais claro e compreensível. Esse processo preserva as proporções originais dos dados e facilita operações subsequentes. A normalização permite que gráficos sejam gerados com dados de diferentes amplitudes sem

distorção visual, melhorando a compreensão do usuário.

A fórmula aplicada para a normalização dos dados é a seguinte:

$$\text{Valor normalizado} = \frac{\text{Valor} - \text{Mínimo}}{\text{Máximo} - \text{Mínimo}}$$

onde:

- *Valor* representa o valor atual do dado a ser normalizado,
- *Mínimo* e *Máximo* correspondem ao menor e maior valor da coluna em questão.

Essa abordagem garante que todos os dados, independentemente de sua escala original, sejam apresentados de forma balanceada e proporcional.

3.5 Interação com os usuários e feedback

Permitir que o usuário interaja com a aplicação é essencial, pois é através dessa interação que se constrói uma experiência mais dinâmica e envolvente. Fornecer *feedbacks* também é fundamental para orientar e informar o usuário durante o uso. Nesta seção, são descritas algumas funcionalidades implementadas para a interação do usuário com a aplicação de forma geral, incluindo o menu principal e suas opções. Além disso, também serão apresentadas as técnicas de feedback aplicadas, como a HUD e a *bounding box*.

3.5.1 Menu principal

O menu principal, de aparência radial e acoplado ao controle esquerdo, tem um papel essencial na aplicação desenvolvida. É através dele que toda a interação primordial acontece. De início, o menu possui quatro botões, cada um com uma função distinta, que podem possuir, ou não, submenus. Abaixo, é possível entender com maior profundidade o papel desempenhado por cada botão do menu principal.

- Interagir com os gráficos: Ao clicar nesse botão, é exibido um submenu com as quatro opções de interação que é possível ter com os gráficos. Ao clicar em um dos botões, ele exibe uma HUD com um texto explicativo sobre como executar cada interação.
- Visualizar um gráfico: Ao ser clicado, esse botão apresenta um submenu com os sete tipos de gráficos que é possível visualizar. Ao selecionar um dos botões, ele alterna a visibilidade do gráfico escolhido e garante que todos os outros estarão invisíveis, dessa forma, só é possível visualizar um gráfico por vez. Além disso, ele também alterna as interações, a fim de permitir que apenas o gráfico atual seja interativo.

- Mudar de ambiente: Esse botão permite que o usuário alterne entre os dois ambientes disponíveis. O padrão é uma sala de controle futurística, mas caso o usuário queira um com menos informações visuais, é possível ir para o outro, que possui apenas um céu noturno e um chão. Ao alternar entre os ambientes, há uma transição suave.
- Redefinir ao estado inicial: Esse botão desfaz todas as modificações feitas na aplicação desde que o usuário entrou na aplicação. Ele retorna os gráficos às suas posições, escala, visibilidade e rotações originais. Além disso, desfaz o que foi feito nas interações de filtrar, retornando as caixas das legendas às suas cores originais, e reativando todos os dados, caso tenham sido desativados pelos filtros. Ele também redefine os menus, retirando o estado de selecionado, caso tenha em algum botão.

No funcionamento do submenu, alguns detalhes foram implementados para um melhor funcionamento. O primeiro deles é que dentro de cada menu, seja o principal ou um submenu, apenas um botão pode estar ativo por vez. Ou seja, apenas um botão do menu principal e um botão de cada submenu por vez pode estar selecionado. Outro detalhe é que caso o usuário alterne entre os botões do menu principal enquanto tem um botão do submenu ativo, quando ele retornar para esse botão principal, o estado do botão do submenu estará salvo.

Nas figuras 3.13, 3.14 e 3.15 é possível observar, respectivamente, o menu principal, o submenu do botão de interações e o submenu do botão de visualizar os gráficos. Os botões de mudança de ambiente e de retorno ao estado inicial não possuem submenus.



Figura 3.13: Menu principal
Fonte: Autoria própria.



Figura 3.14: Menu principal com submenu dos gráficos ativo
Fonte: Autoria própria.

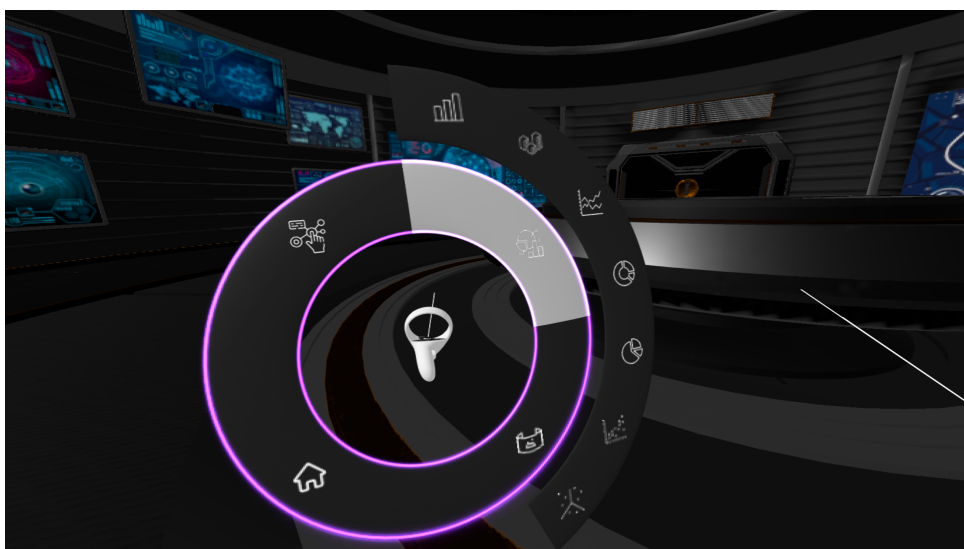


Figura 3.15: Menu principal com submenu de interações ativo
Fonte: Autoria própria.

Na Figura 3.16 é possível notar uma técnica de *feedback* para os usuários aplicada através da mudança de cores com base no estado atual de cada botão. Cada botão tem um código de cor padrão para o estado neutro/não selecionado. Quando há intersecção do controle com o botão, ele entra no estado de *hover* e fica ligeiramente mais claro enquanto a intersecção se mantiver. Quando o botão é selecionado, ele se torna ainda mais claro e permanece nessa cor, até que volte ao estado neutro ou entre no modo temporário de *hover* novamente.

Além disso, ao selecionar algum botão de um dos submenus, o menu radial fica invisível, de maneira que o usuário possa focar totalmente no gráfico ou na mensagem exibida sem ter a visão ocluída. Ele pode alternar a visibilidade do menu a qualquer

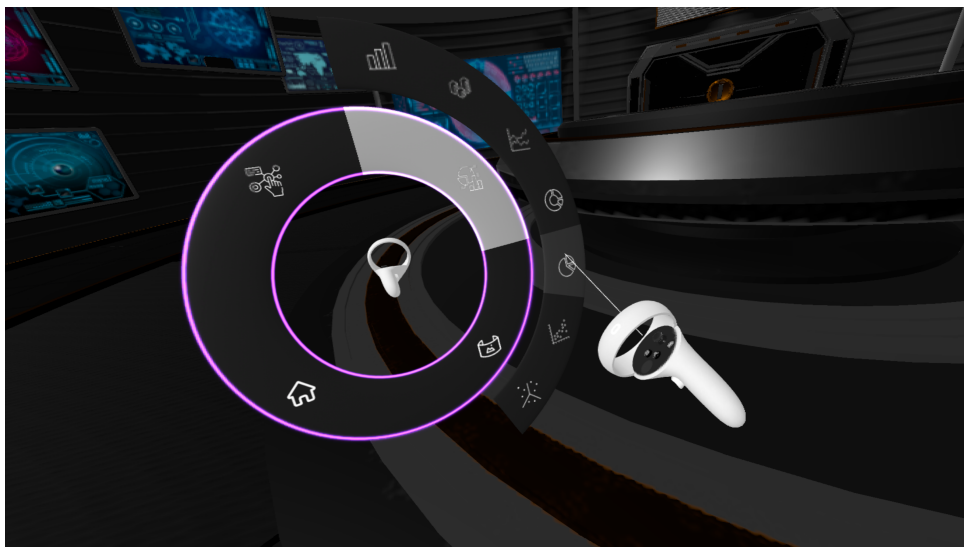


Figura 3.16: Menu principal com estado de hover no submenu ativo
Fonte: Autoria própria.

momento ao pressionar o botão *grip* do controle esquerdo, o mesmo que o menu está atribuído.

3.5.2 HUD

HUDs, sigla em inglês para *Heads-Up Display*, é um termo comumente utilizado em jogos eletrônicos para se referir a elementos gráficos exibidos na tela do usuário, que geralmente apresentam informações cruciais sobre o que está acontecendo. No contexto da aplicação, elas foram utilizadas para exibir instruções aos usuários, no entanto, se limitaram à exibição de mensagens informativas sobre como interagir com os gráficos. Nas figuras 3.17, 3.18, 3.19 e 3.20 é possível conferir as quatro mensagens elaboradas.

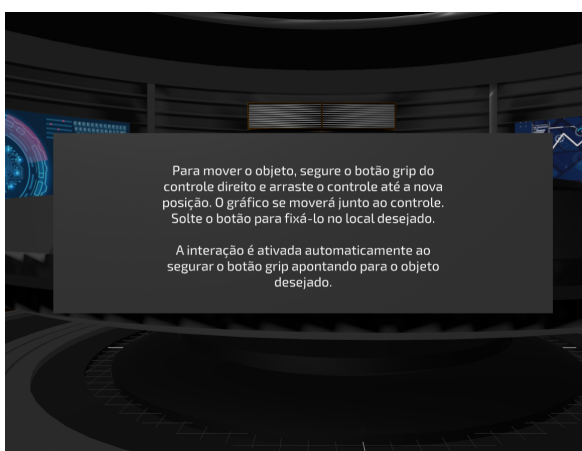


Figura 3.17: HUD explicativa para mover
Fonte: Autoria própria.

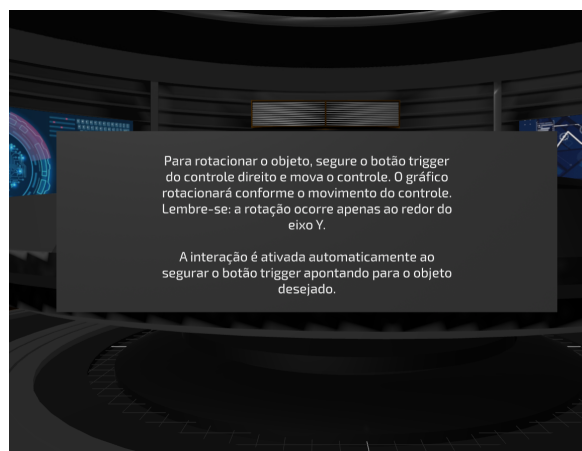


Figura 3.18: HUD explicativa para rotacionar
Fonte: Autoria própria.

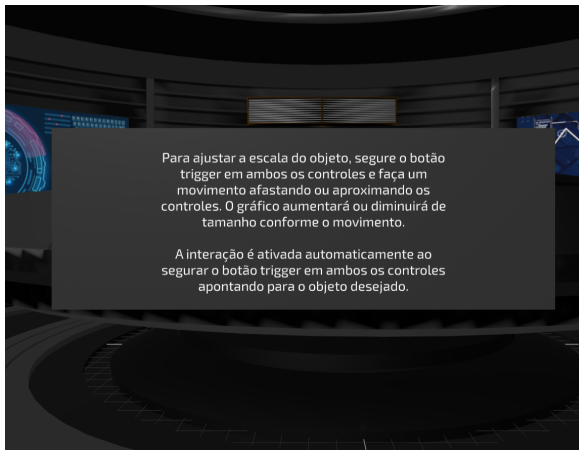


Figura 3.19: HUD explicativa para redimensionar

Fonte: Autoria própria.



Figura 3.20: HUD explicativa para filtrar
Fonte: Autoria própria.

Essas mensagens aparecem quando o botão associado a cada uma delas é selecionado no submenu das interações e desaparecem automaticamente após 15 segundos. É possível escondê-las antes do tempo pré-definido ou reexibi-las. Além disso, as HUDs são fixadas ao usuário e se movem junto a ele.

3.5.3 Bounding Box

Uma *bounding box*, também conhecida como caixa delimitadora, é um objeto que envolve outro de área menor. Na aplicação, essa caixa foi utilizada para verificar a colisão do *raycaster* do controle com a entidade e, assim, permitir a interação com os gráficos. A fim de garantir que se adaptem aos gráficos, foram desenvolvidos dois tipos de *bounding boxes*, uma em formato de cubo, usando o *a-box* do *a-frame*, e outra usando um cilindro, com *a-cylinder*. O cubo é utilizado nos gráficos de barras, linhas e dispersão, enquanto o cilindro é utilizado nos gráficos de pizza e donuts. Nas figuras 3.21 e 3.22 é possível observá-las.

A necessidade de usar a caixa para controlar as interações, ao invés de fazer diretamente na entidade do gráfico, está no controle dos atributos das interações de maneira independente, como, por exemplo, é possível alternar a visibilidade da caixa, sem alternar a visibilidade do gráfico. Isso acontece, pois a caixa delimitadora é uma entidade “irmã”. O diagrama 3.23 ilustra bem a hierarquia das entidades citadas.

Para entender melhor, o gráfico é criado como uma entidade, possuindo os dados, os eixos e as legendas como filhos. Em seguida, ao atribuir o componente de *bounding-box*, são criados uma entidade pai e um filho, que será irmão do gráfico, e funcionará como a *bounding box*. O motivo para não criar a *bounding-box* diretamente como pai é para poder alterar a sua visibilidade sem necessariamente alterar o gráfico. Além disso, a criação da

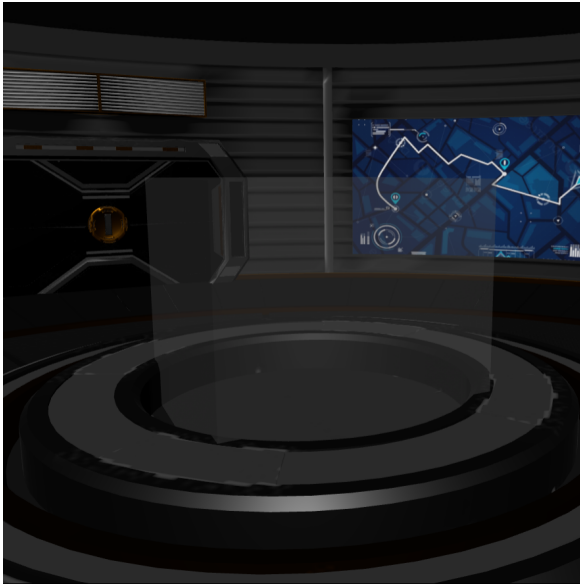


Figura 3.21: Bounding Box de cubo
Fonte: Autoria própria.

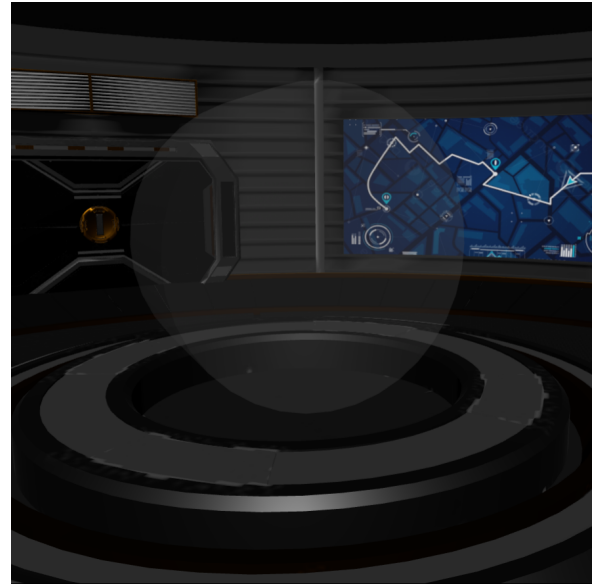


Figura 3.22: Bounding Box de cilindro
Fonte: Autoria própria.

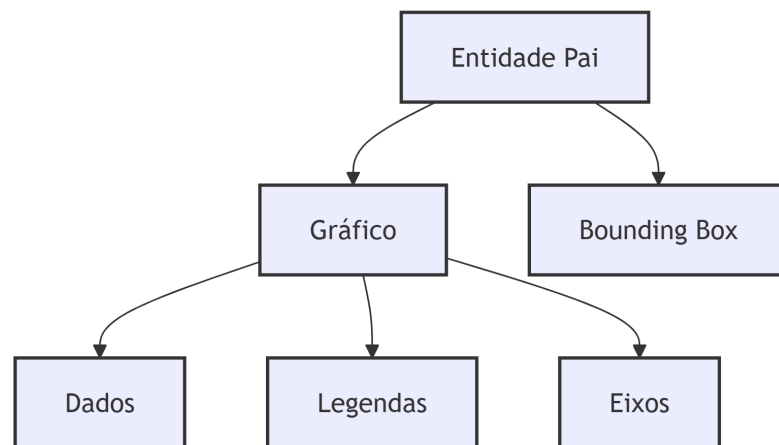


Figura 3.23: Diagrama de hierarquia entre as entidades
Fonte: Autoria própria

entidade pai é responsável por controlar, ao mesmo tempo, o gráfico e a caixa. Dessa forma, as dimensões pessoais da caixa não influenciam no gráfico, e é possível mudar a escala, rotação e posição dos dois, respeitando as suas individualidades.

Quando ocorre interação entre o *raycast* e a caixa, ela assume uma coloração cinza com opacidade baixa. Quando a colisão não está mais ativa, a caixa fica invisível. Essa alteração de visibilidade serve como um poderoso sistema de *feedback* para o usuário, pois, dessa forma, ele consegue saber quando a interação está ativa. Sendo assim, a *bounding-box* tem o objetivo de permitir as interações de posição, rotação e escala, mas para que o usuário possa realizá-las, é necessário apontar para a área o gráfico está localizado, receber o *feedback* de que ela está visível e, então, interagir da maneira que desejar.

3.6 Gráficos Implementados

Como dito anteriormente, todos os gráficos foram implementados usando a metodologia ECS e o framework A-frame. Com isso, cada gráfico possui o seu próprio componente e sistema. Para a utilização dos componentes, só é necessário informar o nome do componente como atributo de uma entidade, passando parâmetros de configuração. O único parâmetro obrigatório é o nome do arquivo de texto, que deve estar na pasta correta no diretório do projeto.

Outras características são opcionais, como, por exemplo, caso um arquivo de texto possua mais colunas do que se é capaz de plotar em um gráfico existente, é possível escolher quais serão plotadas, ao invés do padrão que são as primeiras colunas, além disso, a ordem especificada será respectivamente os eixos x, y e z, então também é possível configurar isso. Também é possível escolher a quantidade de linhas cujos dados serão plotados e esse limite serve para arquivos grandes, no qual se deseja ter apenas uma amostragem. No entanto, as linhas são selecionadas de acordo com a ordem no CSV, no qual, apenas as X primeiras serão escolhidas. A posição e o tamanho de cada dado também podem ser alterados individualmente para cada gráfico.

Outra coisa, é a dimensionalidade dos gráficos, que é definida automaticamente de acordo com a quantidade de colunas escolhidas. Por exemplo, nos gráficos de barras e dispersão, que permitem gráficos com 3 ou 4 dimensões, caso o usuário escolha 3 colunas, ele será um gráfico de 2 eixos e 3 dimensionalidades, pois a terceira coluna serve para categorizar e ser exibida através da cor do dado. Dessa forma, um gráfico com 4 colunas escolhidas no arquivo de texto será exibido com 2 eixos que apresentam ranges de valores e uma legenda que informa outro valor com base na cor dos dados, e ele possuirá dimensionalidade de nível 3.

As cores dos dados são escolhidas por meio de um método auxiliar que retorna uma cor no padrão HSL. Ele calcula a normalização do valor do dado em relação ao valor mínimo e máximo daquele gráfico, e utiliza esse valor normalizado como base para o cálculo do matiz. Os valores de valor e saturação são fixos. Isso permite criar uma paleta de cores semelhante para todos os gráficos, ao mesmo tempo que permite que os dados tenham cores diferentes entre si.

Por fim, é importante destacar que todos os componentes gráficos foram desenvolvidos com base em lógica matemática própria para a geração das representações visuais. Nenhuma biblioteca adicional foi utilizada, garantindo que o processamento e a renderização dos gráficos fossem realizados de maneira customizada, o que proporciona maior controle sobre o comportamento e a performance da aplicação.

3.6.1 Gráfico de Barras

Os gráficos de barras são altamente populares, sendo utilizados principalmente quando se deseja visualizar dados que se distribuem em diferentes categorias. Eles representam os dados usando barras retangulares e é frequentemente empregado para detectar tendências em conjuntos de dados categóricos. O gráfico é formado a partir de várias barras, no qual cada uma representa uma linha do arquivo de texto. Na aplicação desenvolvida, elas foram criadas usando a primitiva **a-box** disponibilizada pelo *framework* A-frame e possui as propriedades: profundidade, largura, altura, posição, rotação e nível de característica metálica.

Esse tipo de gráfico foi implementado para permitir a visualização com 2 ou 3 eixos, a depender da quantidade de colunas recebidas de entrada. E a disposição das barras no gráfico representa esses valores, seja no espaço bi ou tridimensional. As versões iniciais durante a implementação estão exibidas nas Figuras 3.24 e 3.25.

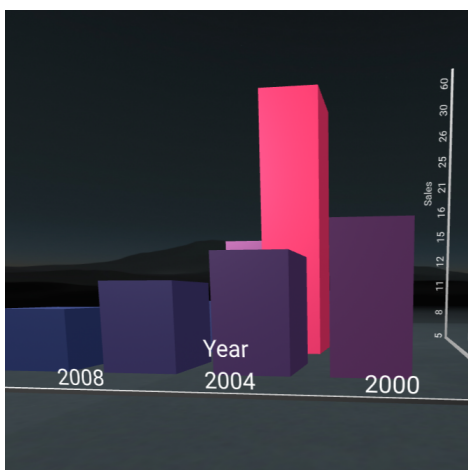


Figura 3.24: Gráfico de barras com 3 dimensões

Fonte: Autoria própria

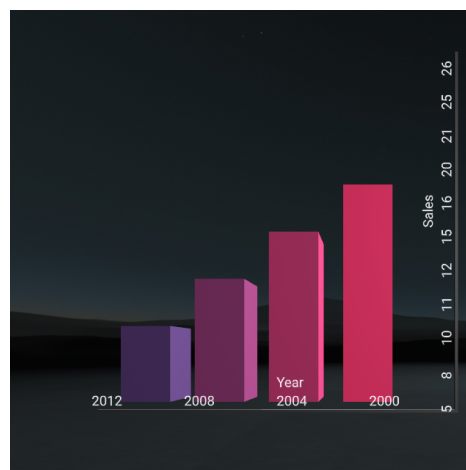


Figura 3.25: Gráfico de barras com 4 dimensões

Fonte: Autoria própria

3.6.2 Gráfico de Dispersão

Os gráficos de dispersão são caracterizados por uma coleção de pontos no espaço, usados principalmente para visualizar a relação entre duas variáveis, através da distância entre os dados. Esse gráfico foi implementado utilizando a primitiva **a-sphere** do a-frame, no qual cada esfera representa um dado no gráfico e sua posição no eixo correspondente. Para cada esfera é possível alterar os atributos: raio, posição, cor e metalidade.

Assim como no gráfico de barras, esse gráfico também foi projetado para permitir a visualização com 2 ou 3 eixos, a depender da quantidade de colunas recebidas de entrada. As versões iniciais durante a implementação estão exibidas nas Figuras 3.26 e 3.27.

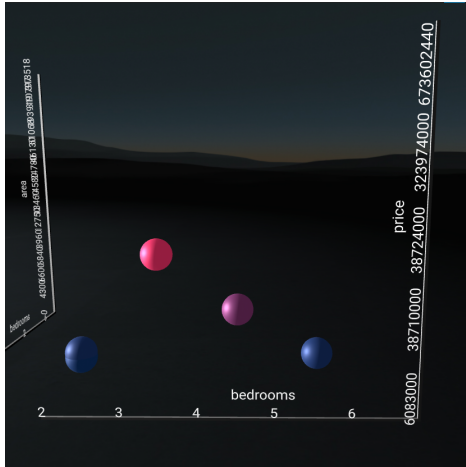


Figura 3.26: Gráfico de dispersão com 3 dimensões

Fonte: Autoria própria

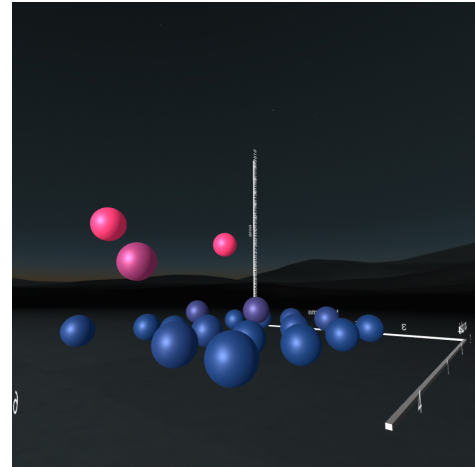


Figura 3.27: Gráfico de barras com 4 dimensões

Fonte: Autoria própria

3.6.3 Gráfico de Pizza

O gráfico de pizza, também conhecido como gráfico de setores, tem como objetivo, sobretudo, mostrar informações de distribuição sobre um todo. Ele facilita a visualização e a compreensão das partes proporcionais de um conjunto de dados. É um dos gráficos mais utilizados e por isso foi considerado essencial para a aplicação.

Além disso, pensando em trabalhos futuros, há bastante potencial em adicionar dimensões extras, por meio de animação, transparência, textura ou tamanho (largura, altura ou raio) e transformar gráficos mais simples e comuns em gráficos mais interessantes e completos por meio de um ambiente imersivo interativo.

Para implementar esse gráfico, foi utilizada a primitiva `a-cylinder` para compor cada “fatia” do gráfico. Atributos como raio, ângulo total e ângulo de início foram essenciais para que a construção dos gráficos tivesse sucesso. Usar ângulos possibilitou controlar de forma automática o posicionamento, a rotação e a construção dos gráficos independente da quantidade de dados e do tamanho de cada uma das suas partes. É possível visualizar a versão inicial durante o período de implementação na Figura 3.28.

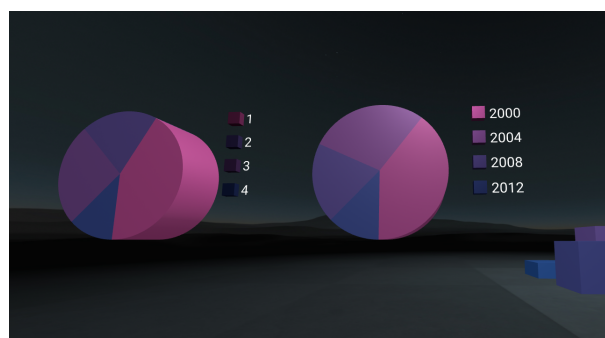


Figura 3.28: Gráficos de pizza

Fonte: Autoria própria

3.6.4 Gráfico de Donuts

O gráfico de donuts se caracteriza por representar os dados como partes proporcionais de um disco com um espaço vazado no meio. Ele é bastante semelhante ao gráfico de pizza, com a principal diferença sendo o aspecto visual. No entanto, pode apresentar uma melhor legibilidade em alguns casos. Por ter uma aparência semelhante, a implementação foi feita com base no componente do gráfico anterior, mudando apenas alguns atributos básicos do a-frame. Enquanto de pizza utilizou a primitiva de um cilindro, o gráfico de donuts utilizou a de um toroide. Atributos como raio interno, raio tubular e tamanho do arco foram essenciais na criação. A versão inicial durante o período de implementação está apresentada na Figura 3.28.

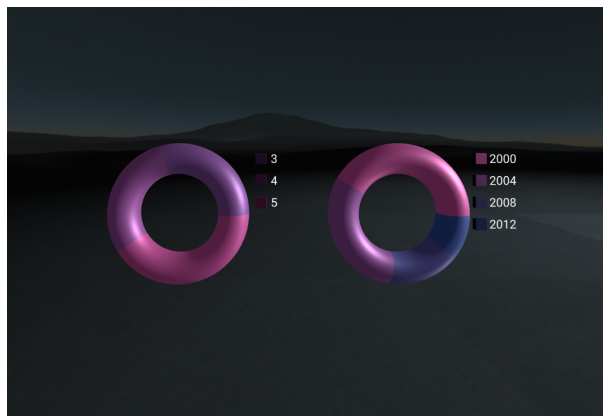


Figura 3.29: Gráfico de donuts
Fonte: Autoria própria

3.6.5 Gráfico de Linhas

O gráfico de linhas apresenta a relação entre dados através da conexão entre eles por meio de uma linha reta. Ele é comumente usado para representar uma série de pontos ao longo do tempo. Para um gráfico de linhas, existem duas possibilidades: o uso de dados contínuos e o uso de dados discretos. Quando os dados são contínuos, eles possuem uma continuidade natural entre si. A implementação consiste em criar uma representação visual para cada um dos dados e eles se ligarão, formando uma linha fluida semelhante a uma série temporal.

No entanto, a implementação desse tipo de gráfico no presente trabalho considerou como entrada dados discretos, que não possuem a continuidade natural como uma propriedade intrínseca. Nesse caso, a linha deve conectar os pontos para facilitar a visualização de alguma tendência ou padrão entre os valores distintos. Dessa forma, uma relação entre os dados precisa ser gerada e, para isso, uma das colunas deve ser escolhida para servir como atributo de relação. Além disso, também foi necessário criar uma interpolação entre esses pontos a fim de preencher as lacunas existentes.

Usando A-frame

No A-frame não há nenhuma primitiva para criação de linhas, portanto, duas opções foram consideradas. A primeira delas consistia em criar um cubo para cada posição original, e estender e rotacionar cada um deles até que se completassem. A segunda ideia tinha como metodologia a criação de vários cubos para preencher o espaço entre um dado e outro. Ambos tinham como objetivo o preenchimento da lacuna existente entre os pontos originais, de forma que o resultado se assemelhasse a uma linha contínua.

A opção escolhida foi a segunda e, para isso, foi necessário gerar um cubo para cada ponto da interpolação. O ideal era gerar a mínima quantidade possível de cubos que desse a impressão de uma linha. No entanto, para que ficasse minimamente uniforme, a quantidade de cubos ainda foi consideravelmente grande. O resultado desse gráfico, no entanto, não foi nada satisfatório em termos de desempenho, tendo em vista que cada entidade gerada e adicionada a cena gera um custo de processamento. Quando testado no Meta Quest 2, o FPS obtido foi baixo. A figura 3.30 mostra o resultado do gráfico desenvolvido.

Além disso, também é possível perceber que o resultado não agradou em termos visuais. Mesmo com muitos cubos em cada linha, ela não ficou com aparência fluida, tendo em vista que dá para perceber claramente onde um cubo começa e termina. Para melhorar visualmente, o desempenho seria ainda mais sacrificado ao continuar aumentando a quantidade de cubos entre eles, até que o espaço entre um e outro fosse mínimo.

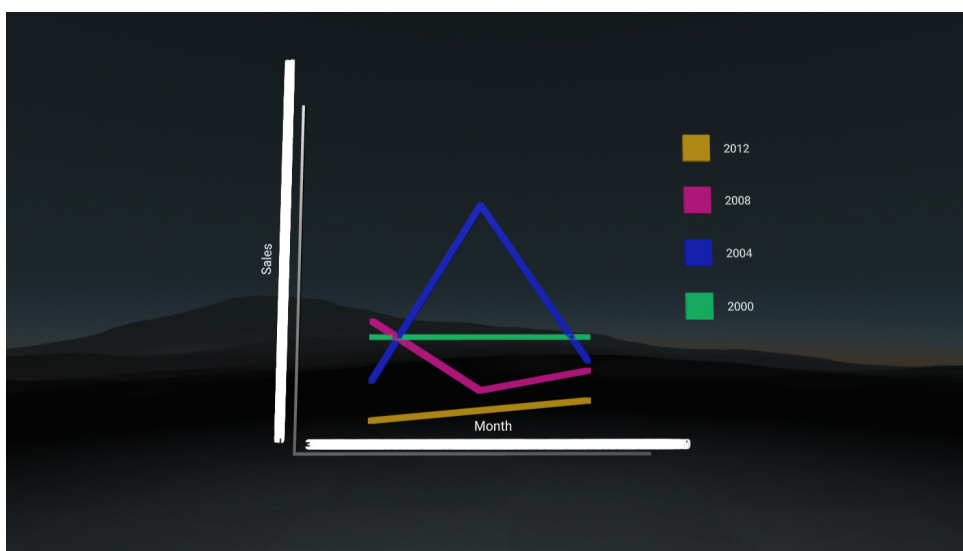


Figura 3.30: Gráfico de linhas usando A-frame

Fonte: Autoria própria

Usando Three.js

A metodologia adotada com o A-frame não apresentou os resultados esperados e, portanto, foi decidido ir um pouco mais abaixo no nível de implementação e testar o uso do Three.js. Tendo em vista que o a-frame foi criado com base nesse último citado, o *framework* não possui todos os componentes, no entanto, dá acesso total à API original.

Para essa implementação, inicialmente, a geração das linhas foi realizada criando vértices e índices, que formam uma *BufferGeometry* para representar os dados em 3D. Para cada ponto da linha, foram definidos oito vértices que formam uma pequena caixa 3D, dando volume à linha no espaço, em vez de ser apenas uma linha bidimensional. Os vértices foram organizados em grupos de quatro, correspondendo às extremidades das faces da caixa ao redor do ponto. Em seguida, os índices foram utilizados para conectar esses vértices, criando faces entre os pontos consecutivos e formando segmentos sólidos da linha. A *BufferGeometry* foi então construída com esses vértices e índices, e aplicada a um material, resultando em uma linha tridimensional visível de todos os ângulos na cena.

Além disso, foi decidido representar o gráfico em quatro dimensões, ao invés de três, como havia sido feito anteriormente. O resultado dessa versão do gráfico de linhas pode ser observado na Figura 3.31.

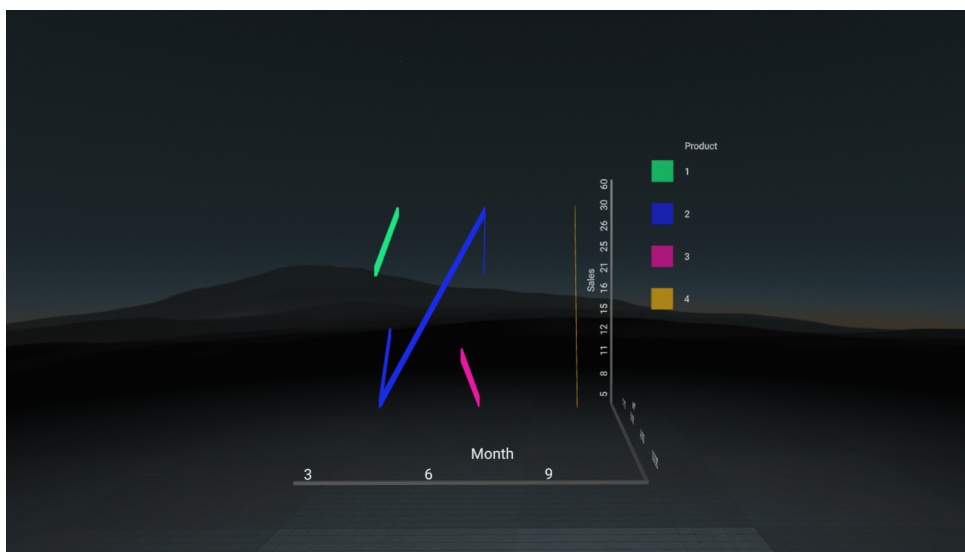


Figura 3.31: Gráfico de linhas usando ThreeJS, versão inicial
Fonte: Autoria própria

É possível perceber o formato quadriculado das mesmas. Além disso, a linha se torna invisível de alguns ângulos. Portanto, foi desenvolvida uma terceira variação do gráfico de linhas. Dessa vez, porém, foram utilizados os componentes de *CatmullRomCurve3* e *TubeGeometry* do Three.js. O primeiro recebe a lista de pontos originais e é responsável por gerar uma curva que passa por eles no espaço 3D, produzindo uma trajetória contínua e suave. Ela usa a interpolação *Catmull-Rom*, que suaviza a trajetória entre os pontos fornecidos. Após isso, o uso do *TubeGeometry* se torna necessário,

tendo em vista que ele é o responsável por criar uma geometria de tubo em torno dessa curva tridimensional. Esse componente utiliza a curva gerada pelo `CatmullRomCurve3` como caminho central e cria uma malha 3D em formato de tubo ao longo dela, onde cada segmento do tubo segue a orientação e a trajetória da mesma.

Dessa forma, o primeiro é responsável por criar a curva com a interpolação dos dados recebidos, enquanto o segundo envolve essa curva com uma geometria de tubo, gerando a visualização 3D. Com os dois componentes em conjunto, é possível criar o gráfico de linhas de maneira simplificada, no qual o resultado é uma linha 3D no formato de um tubo, que representa os dados no gráfico de maneira visualmente uniforme e suave.

3.6.6 Componentes auxiliares

Quando nos referimos a gráficos, alguns componentes auxiliares são essenciais. No caso desse trabalho, esses elementos são os eixos e as legendas. Responsáveis por fornecer informações sobre o gráfico representado, os textos que ambos exibem são extraídos diretamente das *headers* dos arquivos de texto, o que permite explorar ao máximo as informações contidas nos documentos e transmitir para os usuários de maneira simplificada.

Eixos

Os eixos formam uma moldura ao redor dos gráficos e são extremamente importantes para os mesmos. Eles servem como guias para o entendimento de informações sobre os dados, ao mesmo tempo em que fornecem uma referência para marcadores de medidas e atributos.

No momento da criação de um gráfico, eles são gerados e adicionados automaticamente. As posições deles são fixas e dependem da quantidade de dimensões do gráfico. Cada eixo é formado por um título, uma caixa retangular um pouco maior que o tamanho do gráfico e marcadores distribuídos de maneira uniforme ao longo da reta, de maneira condizente com a posição dos dados. Esses marcadores são ordenados por ordem crescente e estão divididos ao longo do comprimento total, para se adaptar a uma quantidade variada de medidas. O título fica centralizado um pouco acima deles. Na imagem abaixo, é possível observar os eixos para gráficos bi e tridimensionais.

Legenda

As legendas são usadas para explicar as informações apresentadas no gráfico, permitindo que os usuários compreendam facilmente atributos como cor, textura e símbolo. Nos gráficos desse trabalho, o único atributo presente desse tipo é a cor. Portanto, as legendas servem para indicar o que a cor de um dado representa. Importante notar que diferentes dados de um mesmo gráfico podem compartilhar a mesma cor, o que significa que eles compartilham aquela característica.

Por padrão, o atributo usado para separar os grupos definidos na legenda sempre será referente à última coluna escolhida pelo usuário ao instanciar um novo gráfico. Sendo assim, cada grupo de dados que compartilha o mesmo atributo na última coluna recebe a mesma cor. No qual, cada cor é atribuída a uma caixa pequena, com o nome referente a esse atributo compartilhado e com um título principal, retirado da *header* dessa coluna. Dessa forma, as legendas possuem título, caixas com cores fazendo referências aos dados e *label* informando sobre o que cada cor representa.

Além disso, como dito no início dessa seção, os usuários decidem as colunas que serão usadas e isso define a quantidade de dimensões que o gráfico terá, sendo a cor uma delas. Com isso, alguns gráficos podem ter três eixos (x, y e z), mas quatro dimensões, como os gráficos de barras, dispersão e linhas. Eles também podem ter dois eixos (x e y), mas três dimensões, como os gráficos de barras e de dispersão. E, por fim, podem possuir legendas, mas não possuir nenhum eixo, como é o caso dos gráficos de pizza e de donuts. No caso desses últimos, esse componente é ainda mais importante, tendo em vista que não possuem eixos e a legenda é o único indicador de informações do gráfico.

3.7 Interações com os gráficos

A visualização de dados é potencializada quando as interações são exploradas em conjunto. Na realidade estendida, há uma ampla gama de possibilidades, no entanto, ainda é uma área em exploração, sem padrões e boas práticas consolidados. Para essa aplicação, as interações relacionadas à posição, rotação, escala e filtragem foram escolhidas para serem trabalhadas. Elas são realizadas por meio dos controles e ativadas automaticamente ao executar uma sequência de passos. Nesta seção, serão descritos os processos e as técnicas implementadas para torná-los realidade.

3.7.1 Mover

Mover um objeto é uma das interações mais básicas e, por isso, uma das mais importantes. A solução desenvolvida para proporcionar essa interação no ambiente de realidade virtual de maneira fluida e natural foi anexar o objeto ao controle, tornando-o filho hierarquicamente.

O fluxo para executar essa interação está ilustrado na Figura 3.32.

Isso permite que o usuário aproxime, afaste, posicione mais para direita ou esquerda, e concede uma liberdade muito grande.

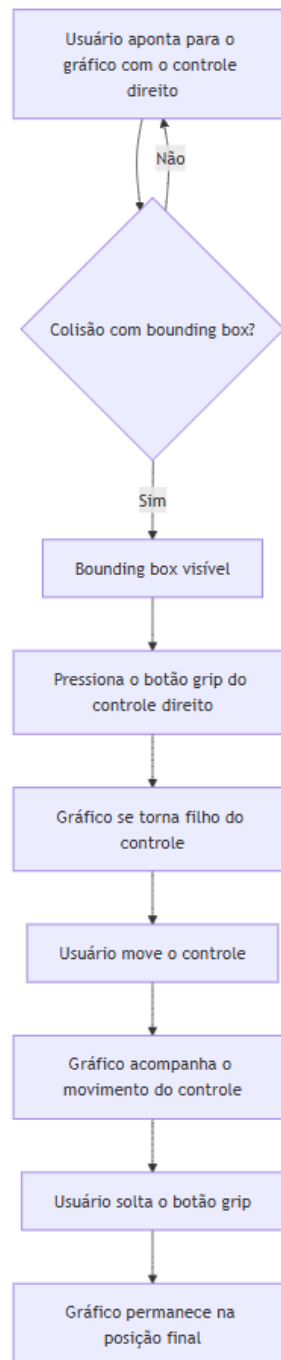


Figura 3.32: Fluxograma do processo de mover um objeto
Fonte: Autoria própria

3.7.2 Rotacionar

Em seguida, foi desenvolvida a interação que permite a rotação dos gráficos. No ambiente virtual, o usuário pode e é incentivado a andar pelo ambiente e ao redor do gráfico a fim de observar todos os ângulos. Ainda assim, rotacionar o gráfico tem o seu valor, visto que o usuário pode visualizar os diferentes ângulos de forma mais fácil e prática, sem alterar sua posição inicial. Além disso, a rotação foi fixada para funcionar

apenas em relação ao eixo Y, a fim de evitar confusões desnecessárias.

Dessa forma, a interação de rotacionar implementada funciona por meio do seguinte fluxo, ilustrado na Figura 3.33.

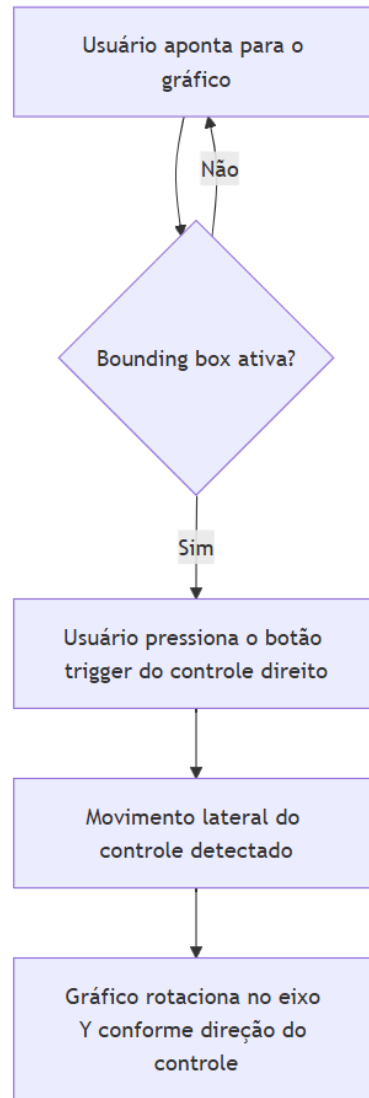


Figura 3.33: Fluxograma do processo de rotacionar um objeto
Fonte: Autoria própria

3.7.3 Redimensionar

O redimensionamento permite que um usuário aumente ou diminua o tamanho do gráfico, de maneira que fique mais confortável e legível. Para um usuário alterar a escala do gráfico, é obrigatório o uso dos dois controles. Dessa forma, ao pressionar os botões trigger dos dois controles ao mesmo tempo, enquanto aponta para o gráfico, a interação é iniciada. Após isso, ele controlará a mudança da escala através da distância entre os controles.

Dessa forma, a interação de redimensionar implementada funciona por meio do seguinte fluxo, ilustrado na Figura 3.34.

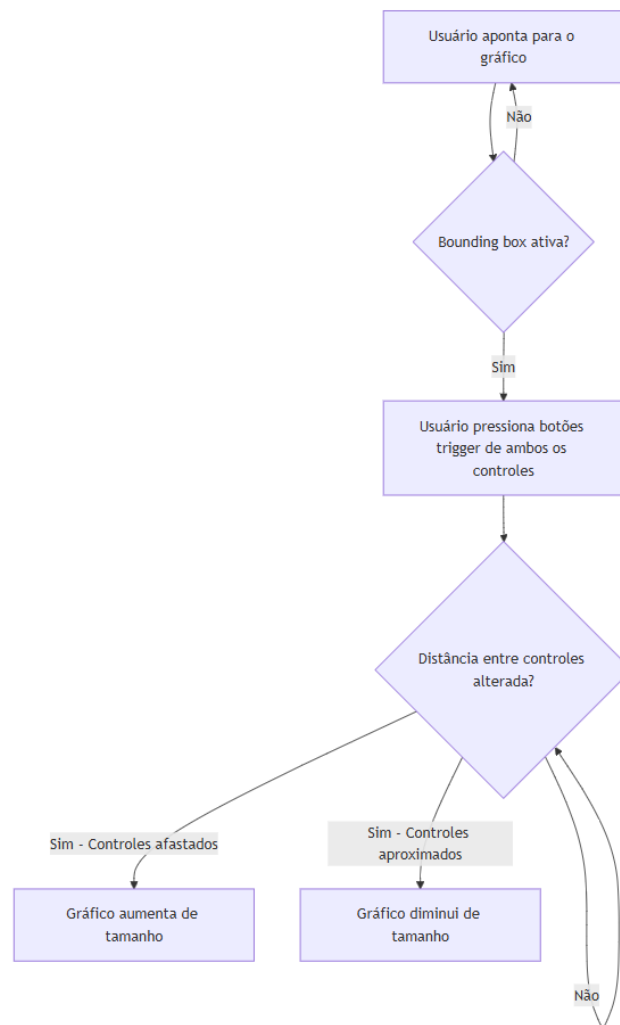


Figura 3.34: Fluxograma do processo de redimensionar um objeto
Fonte: Autoria própria

3.7.4 Filtrar

A interação de filtrar é a que mais difere das previamente citadas, tendo em vista que não altera diretamente uma propriedade do gráfico. No entanto, ela tem uma importância ímpar. A filtragem permite ao usuário ocultar subconjuntos de dados que não são relevantes em determinado momento, facilitando o foco em aspectos específicos e promovendo *insights* mais profundos. A técnica de filtragem implementada nesta aplicação utiliza interações específicas com os controles do dispositivo para alternar a visibilidade dos dados, abrindo a possibilidade para futuros aprimoramentos.

Dessa forma, o processo de filtragem foi implementado de forma a permitir uma experiência de usuário intuitiva, mas eficiente. O fluxo segue os passos ilustrados na

Figura 3.35.

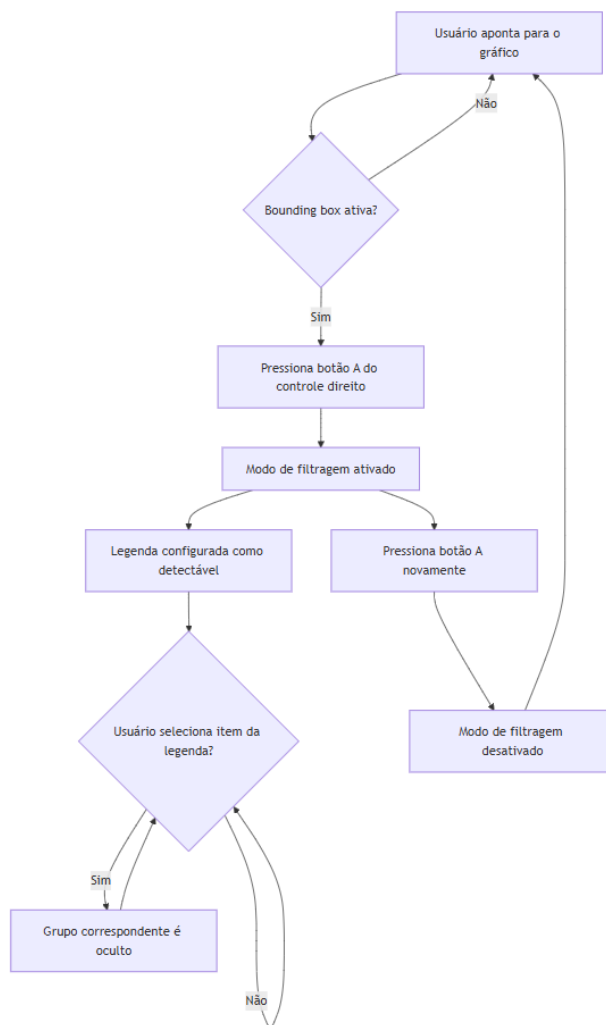


Figura 3.35: Fluxograma do processo de filtrar dados
 Fonte: Autoria própria

3.8 Aparência

A aparência dos ambientes virtuais é essencial para tornar a experiência imersiva e visualmente atraente para o usuário. Na criação desta aplicação, foram aplicadas técnicas de otimização e efeitos visuais, visando oferecer uma ambientação futurista e cativante que complementa a experiência de visualização de dados, mantendo um bom desempenho. As subseções a seguir detalham as escolhas e as técnicas implementadas.

3.8.1 Modelo 3D e otimização

A modelagem 3D em ambientes WebXR requer modelos otimizados para o bom desempenho da aplicação, visto que o processamento gráfico no navegador de óculos de

realidade estendida é limitado em comparação com desktops. Por isso, opta-se geralmente por modelos *low-poly*, com menor quantidade de polígonos, o que reduz o impacto na taxa de quadros e no desempenho geral.

Para esta aplicação, foi utilizado um modelo disponibilizado no *Sketchfab*, intitulado *Sci-Fi Command Room Interior* 3DDisco (2024). Este modelo apresentava uma alta densidade de polígonos e não era otimizado para ambientes VR, resultando em um desempenho ruim. Com um grande número de elementos e detalhes decorativos, o modelo necessitou de um trabalho de otimização. As etapas de simplificação aplicadas foram:

- Remoção de elementos supérfluos: Componentes decorativos, como esferas e outros detalhes que não eram relevantes para a visualização dos dados, foram removidos manualmente. Esta remoção diminuiu substancialmente a contagem de polígonos.
- Redução de vértices e faces manualmente: Elementos como totens centrais, armários e ornamentos nas portas e no teto foram deletados vértice a vértice, reduzindo ainda mais a complexidade do modelo.
- Uso de ferramentas de otimização no blender: Funções automáticas, como o modificador *Decimate*, foram aplicadas para simplificar ainda mais as malhas. Além disso, a função *Merge Vertices* ajudou a eliminar vértices redundantes.

A Figura 3.36 exibe o resultado do cenário principal após as otimizações.

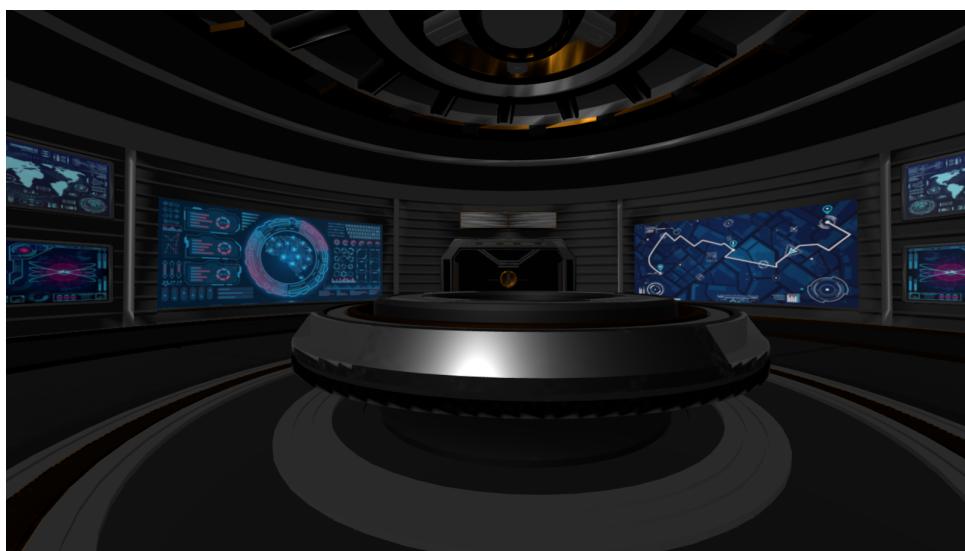


Figura 3.36: Cenário principal
Fonte: Autoria própria

3.8.2 Ambiente secundário

A aplicação oferece a funcionalidade de alternar entre dois ambientes distintos, permitindo ao usuário escolher o contexto visual que mais lhe agrada. O ambiente padrão

é uma sala de controle futurística, com design inspirado em espaços tecnológicos e de alta performance. Como alternativa, existe um ambiente secundário mais minimalista, ideal para uma visualização focada, sem distrações.

O ambiente secundário inclui um chão e uma esfera de *skybox* configurados para manter a sensação de profundidade e proporcionar uma experiência de visualização consistente, adaptando-se ao novo contexto. Para tornar a troca de ambiente mais fluida, foi implementada uma transição suave entre os dois cenários.

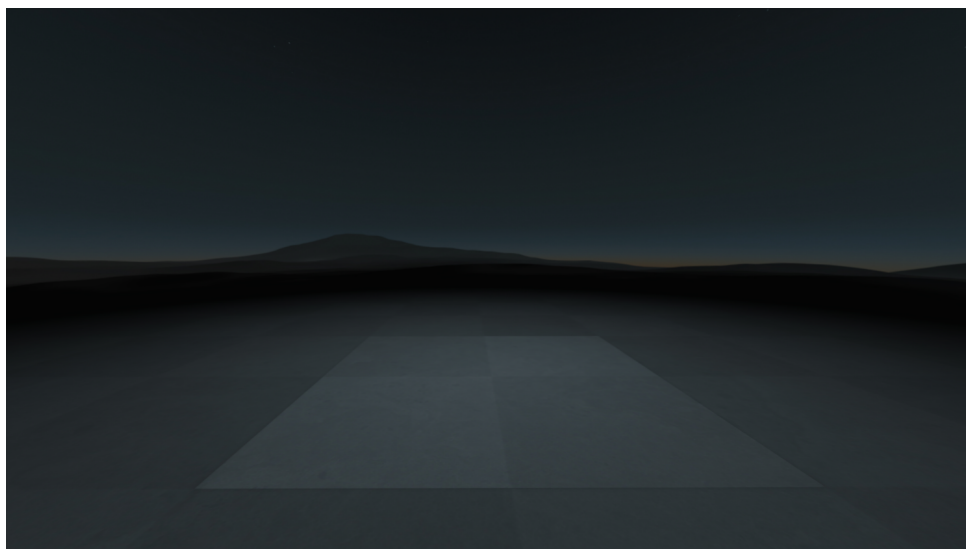


Figura 3.37: Cenário secundário
Fonte: Autoria própria

3.8.3 Efeito neon

Para destacar os gráficos e adicionar um toque futurista, foi adicionado o efeito *Fake Glow*, que simula um brilho suave ao redor dos elementos, dando mais profundidade e destaque visual aos dados. Inicialmente, foram testadas outras técnicas, como os materiais de pós-processamento oferecidos pelo Three.js, porém o *shader* foi o que trouxe o resultado esperado de maneira prática, ao fornecer um visual mais interessante e com uso fácil.

O *shader* utilizado para criar esse efeito foi adaptado do projeto de código aberto desenvolvido por [ektogamat \(2024\)](#). O uso do efeito agrega à estética futurista e causa uma impressão visual mais impactante, tornando os gráficos mais atrativos e incentivando o usuário a interagir com a visualização dos dados.

3.8.4 Animações

Para criar uma experiência inicial mais envolvente, foi implementada uma animação simples de escala que ocorre quando cada gráfico é exibido pela primeira vez.

Nesse processo, cada elemento do gráfico, os eixos e as legendas aumentam gradualmente sua escala de 0 até 1, criando um efeito de entrada suave.

Essa técnica de animação por escala foi escolhida visando chamar atenção para os dados por meio do crescimento progressivo. Ela contribui para uma apresentação visual mais dinâmica e incentiva o usuário a interagir com a visualização dos dados.

Capítulo 4

Resultados

Nesta seção, são apresentados os principais resultados obtidos durante o desenvolvimento da aplicação de visualização de dados em realidade estendida. A análise abrange as dificuldades enfrentadas, funcionalidades implementadas e testes realizados em diferentes dispositivos.

4.1 Dificuldades enfrentadas

Durante o desenvolvimento da aplicação, diversos desafios técnicos surgiram, exigindo soluções e otimizações para garantir a funcionalidade e a usabilidade esperadas. Nesta seção, serão discutidos os principais problemas enfrentados e as abordagens adotadas para superá-los.

4.1.1 FPS e Otimização

O desempenho da aplicação foi um dos maiores desafios no desenvolvimento e também um dos principais pontos de preocupação, dado o impacto direto na experiência do usuário em ambientes imersivos. Dessa forma, para garantir uma taxa de quadros por segundo (FPS) aceitável, o FPS foi observado a cada etapa de desenvolvimento. Os principais fatores que contribuíram para o baixo desempenho foram:

- Como mencionado anteriormente, a primeira versão do gráfico de linhas foi um grande problema em relação ao desempenho. Cada ponto do gráfico era representado por cubos individuais, resultando em milhares de objetos na cena. O FPS médio apenas com esse objeto na cena, era de 8. Isso foi substituído por uma abordagem baseada em linhas renderizadas diretamente, usando o ThreeJS, reduzindo o custo computacional para uma média acima de 40.
- Por causa de uma configuração errada no *raycast*, a verificação de colisões estava ativa para todos os objetos da cena. Uma otimização foi realizada para restringir o

raycasting apenas aos objetos relevantes para a interação do usuário. Essa solução é uma grande recomendação do framework.

- O outro problema de desempenho ocorreu quando houve a inserção do modelo 3D futurista. Como o modelo possuía alta contagem de polígonos, o desempenho foi bastante prejudicado. Para resolver esse problema, o modelo foi otimizado para uma versão mais leve, sem perda significativa de qualidade visual. As técnicas adotadas para essa otimização estão descritas na metodologia. Na Tabela 4.1, são apresentados dados comparativos do modelo antes e depois da otimização.

Parâmetro	Antes da otimização	Depois da otimização
Vértices	374.253	44.113
Faces	124.751	27.159
Triângulos	124.751	27.165
Objetos	8	22

Tabela 4.1: Comparação dos dados do modelo 3D antes e depois da otimização

Em alguns momentos, a aplicação apresentou um desempenho inferior a 4 FPS, considerada insuficiente para interações fluidas. Como resultado, essas técnicas reduziram drasticamente o número de polígonos e, conseqüentemente, melhoraram o desempenho da aplicação.

4.1.2 Problemas com a lógica, eventos e estados

Durante a implementação das interações, ocorreram problemas relacionados aos estados, que ocasionaram *bugs* inesperados. Por exemplo, na interação de filtros, ao desativar todas as interações, nem sempre isso funcionou. E algumas interações eram ativadas mesmo quando não deveriam ser.

Para resolver esses problemas, foi implementado um controle mais rigoroso dos estados dos elementos interativos, modularizando o código e garantindo que o controle dos estados das interações estivesse funcionando corretamente e apenas um gráfico ou objeto pudesse ser manipulado de cada vez.

A implementação de interações exigiu um controle rigoroso dos estados das caixas delimitadoras e filtros, para evitar conflitos. Era essencial garantir que apenas um gráfico fosse interativo por vez, evitando situações onde múltiplas interações acontecessem simultaneamente e gerassem comportamentos inesperados.

A abordagem adotada foi o uso de um sistema de gerenciamento de estados, onde cada gráfico ou elemento interativo tinha sua própria lógica de ativação e desativação. Isso assegurou que as interações fossem intuitivas e previsíveis para os usuários.

Essas dificuldades, embora desafiadoras, foram fundamentais para o aprendizado e evolução da aplicação, resultando em um produto final mais robusto e eficiente.

4.1.3 Lógica matemática para gerar os gráficos

Tendo em vista que toda a lógica de implementação dos gráficos foi desenvolvida nesse projeto, sem o uso de bibliotecas externas, isso somou uma dificuldade relativa. A criação de gráficos, exigiu o desenvolvimento de uma lógica matemática precisa para traduzir os dados em representações visuais coerentes. Isso incluiu:

- Garantir que a relação entre os dados estivesse correta, principalmente na interação de filtrar. Sendo necessário filtrar pelas colunas salvas e agrupar os dados corretamente.
- Especialmente para o gráfico de linhas, a interpolação desenvolvida inicialmente teve uma complexidade alta. Além disso, foi necessário testar diferentes implementações a fim de garantir uma solução otimizada e funcional.
- Transferir os dados para representações visuais com posições nos eixos que correspondessem a realidade.
- Um problema recorrente no desenvolvimento dos gráficos foi a sobreposição de dados no mesmo ponto entre os eixos, especialmente em situações onde múltiplos valores compartilhavam a mesma coordenada. Isso prejudicava a legibilidade e dificultava a análise visual. Para lidar com isso, foram aplicadas soluções como a soma de elementos que compartilhavam as mesmas coordenadas, a fim de mostrar apenas a soma/resultado final.

4.1.4 Textos ocultos

No desenvolvimento de gráficos com muitas informações, textos associados a elementos frequentemente se sobrepunham ou ficavam ocultos. A solução foi criar um componente para gerenciar as camadas de texto e gráficos, garantindo que as informações prioritárias fossem exibidas de maneira clara e sem obstruções.

4.2 Aplicação Desenvolvida

A aplicação desenvolvida visa explorar as capacidades de visualização interativa em AR e VR, oferecendo suporte para dispositivos modernos como o Meta Quest 2 e 3. A seguir, são apresentados os principais recursos da aplicação e exemplos de sua interface e funcionamento.

4.2.1 Funcionalidades

O produto final consiste em uma aplicação web compatível com realidade aumentada e virtual. Entre as principais funcionalidades implementadas, destacam-se:

- Visualização de gráficos interativos em AR e VR.
- Transições suaves entre os modos de realidade aumentada e virtual.
- Suporte à manipulação de gráficos através de controles.
- Cinco tipos de gráficos: barras, pizza, donuts, dispersão e linha. Representações em 3D e 4D nos gráficos de barras e dispersão.
- Quatro tipos de interações com os gráficos: mover, rotacionar, redimensionar e filtrar.
- Menu inovador para navegação intuitiva.
- Mensagens de instrução para facilitar o uso.

As imagens que ilustram os resultados, incluindo os gráficos, interações, cenários e detalhes da interface, estão apresentadas ao longo das subsubseções abaixo.

Menu de Interações

A Figura 4.1 exibe o menu principal com o botão de interações ativo e o botão de filtrar no estado de *hover*.

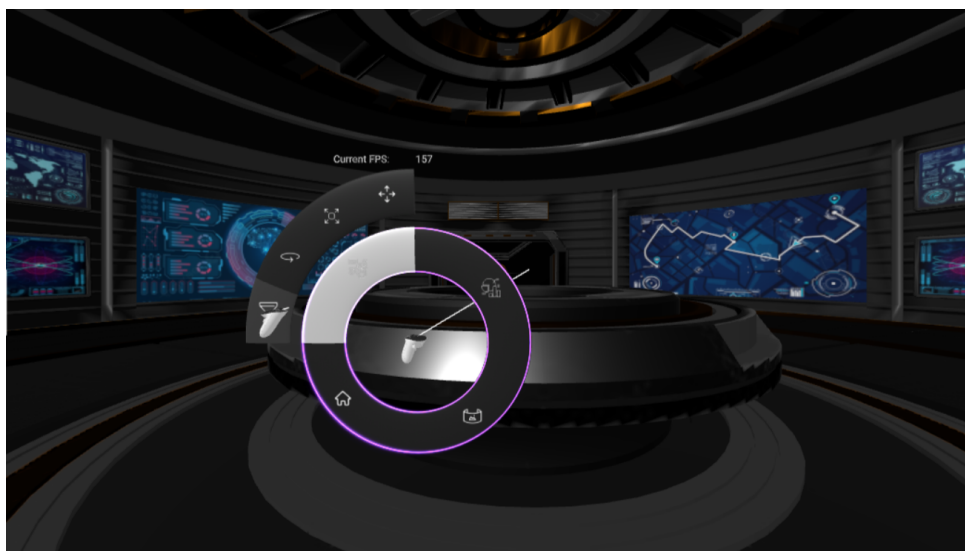


Figura 4.1: Menu de interações
Fonte: Autoria própria

Gráfico de Barras

A Figura 4.2 apresenta um gráfico de barras com três dimensões, enquanto a Figura 4.3 mostra a mesma estrutura, porém com quatro dimensões.

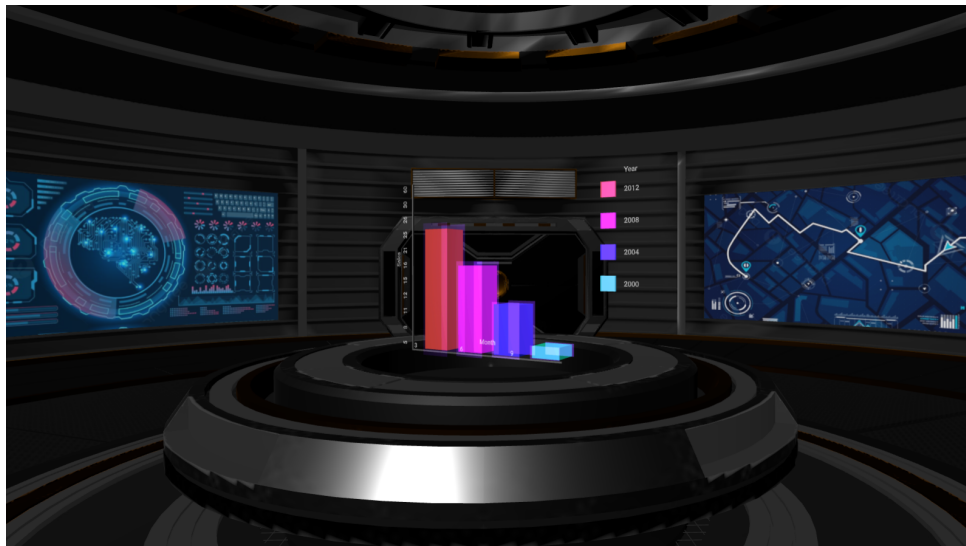


Figura 4.2: Gráfico de barras com 3 dimensões
Fonte: Autoria própria

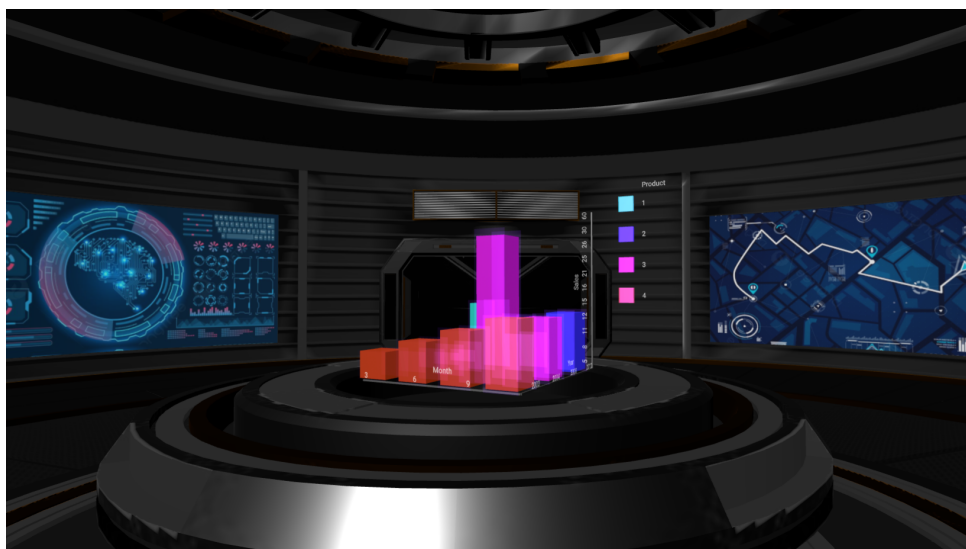


Figura 4.3: Gráfico de barras com 4 dimensões
Fonte: Autoria própria

Gráfico de Dispersão

A Figura 4.4 apresenta um gráfico de dispersão com três dimensões, enquanto a Figura 4.5 mostra com quatro dimensões.

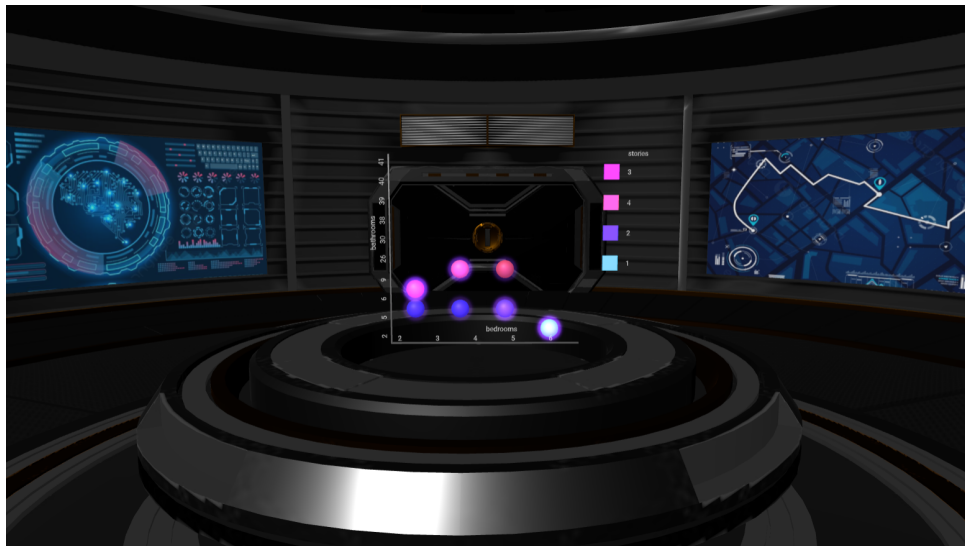


Figura 4.4: Gráfico de dispersão com 3 dimensões
Fonte: Autoria própria

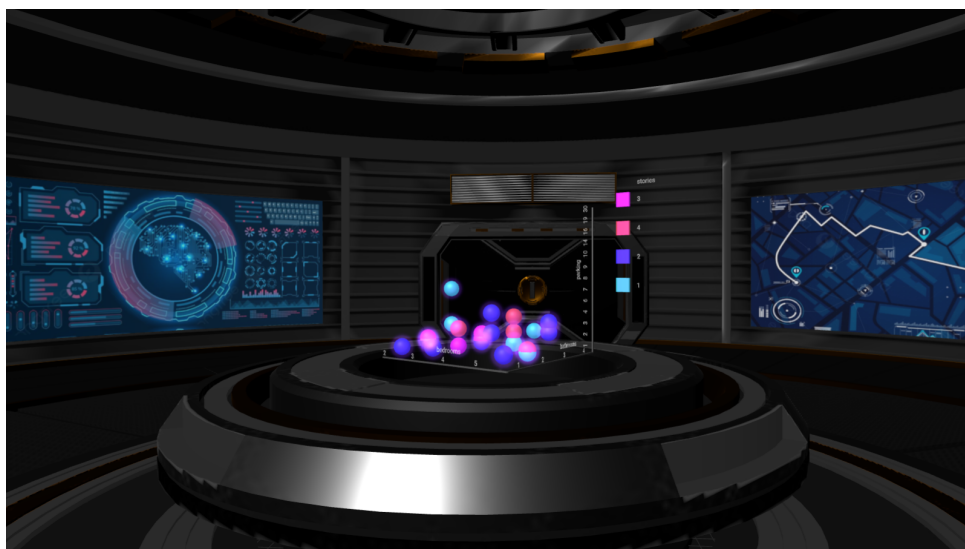


Figura 4.5: Gráfico de dispersão com 4 dimensões
Fonte: Autoria própria

Gráfico de Pizza

A Figura 4.6 apresenta um gráfico de pizza.

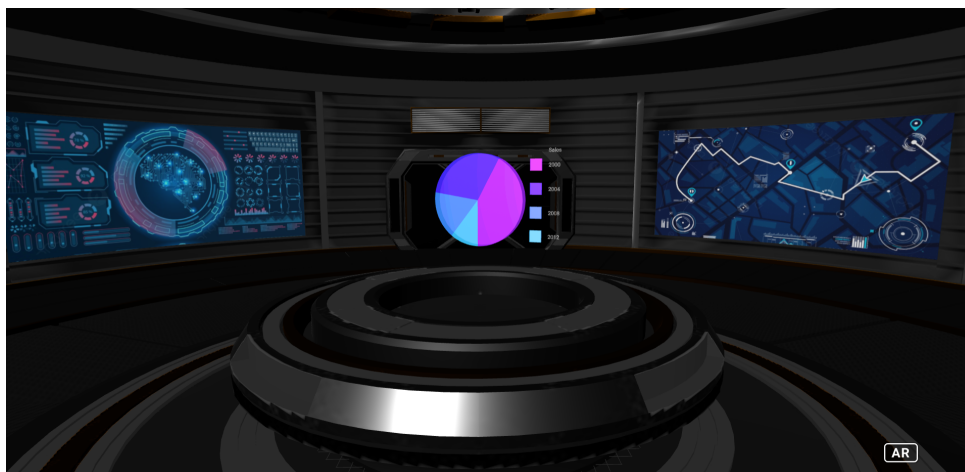


Figura 4.6: Gráfico de pizza
Fonte: Autoria própria

Gráfico de Donuts

A Figura 4.7 apresenta um gráfico de donuts.

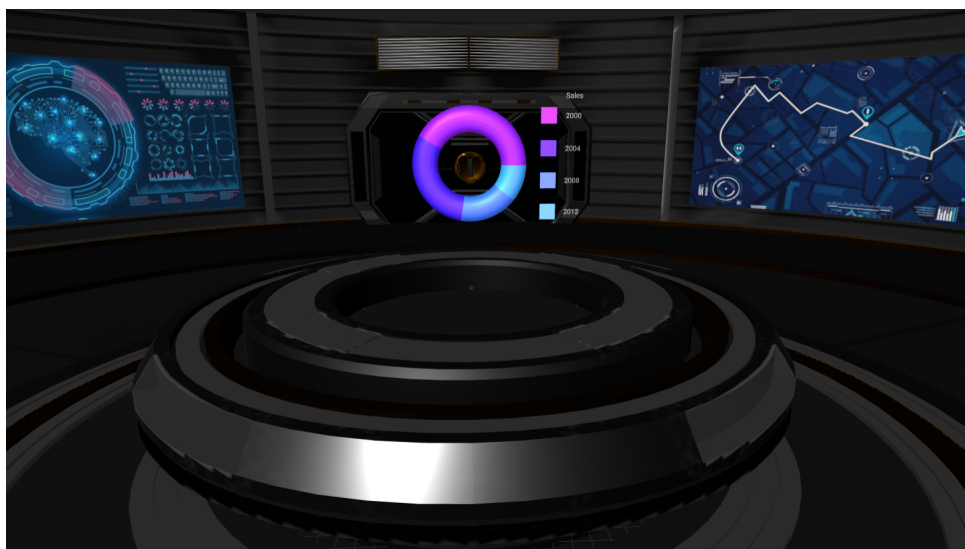


Figura 4.7: Gráfico de donuts
Fonte: Autoria própria

Gráfico de Linhas

A Figura 4.8 apresenta um gráfico de linhas.

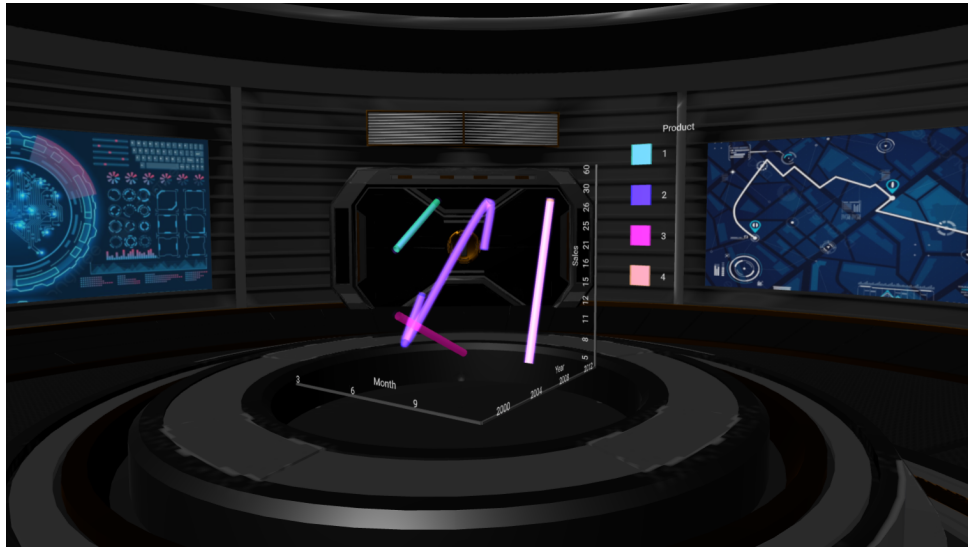


Figura 4.8: Gráfico de linhas

Fonte: Autoria própria

Eixos

A Figura 4.9 apresenta os eixos com duas dimensões, enquanto a Figura 4.10 mostra com três dimensões.



Figura 4.9: Eixos com x e y

Fonte: Autoria própria



Figura 4.10: Eixos com x, y e z
Fonte: Autoria própria

Legenda

A Figura 4.11 apresenta as legendas.

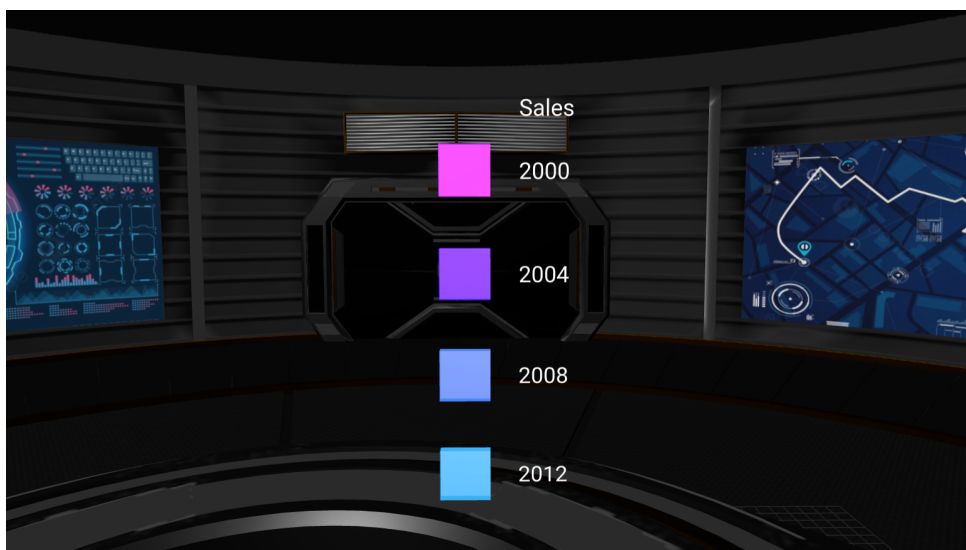


Figura 4.11: Legenda de um gráfico
Fonte: Autoria própria

Interação de Mover

A Figura 4.12 ilustra um gráfico após sofrer alteração na sua posição.

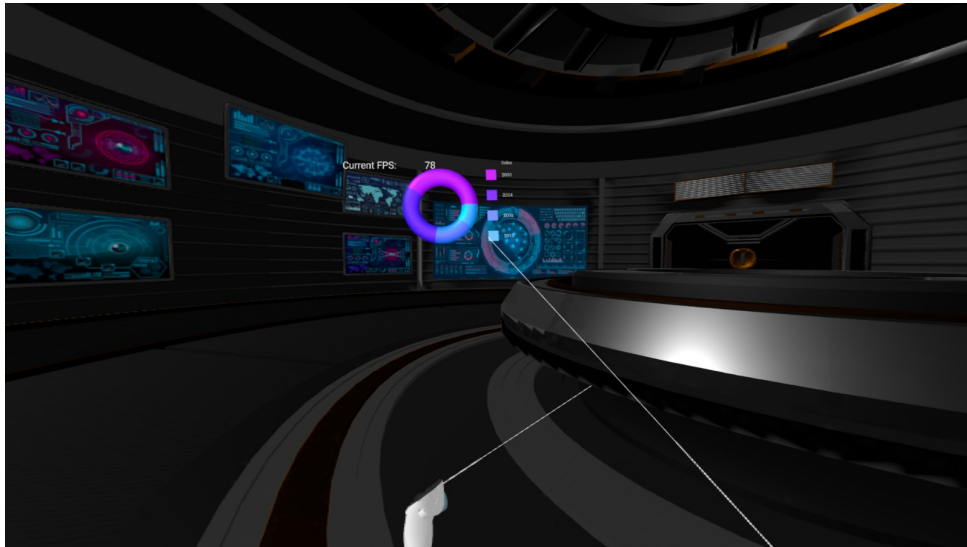


Figura 4.12: Interação de mover ativa.
Fonte: Autoria própria

Interação de Rotacionar

A Figura 4.13 ilustra um gráfico após sofrer alteração na sua rotação.

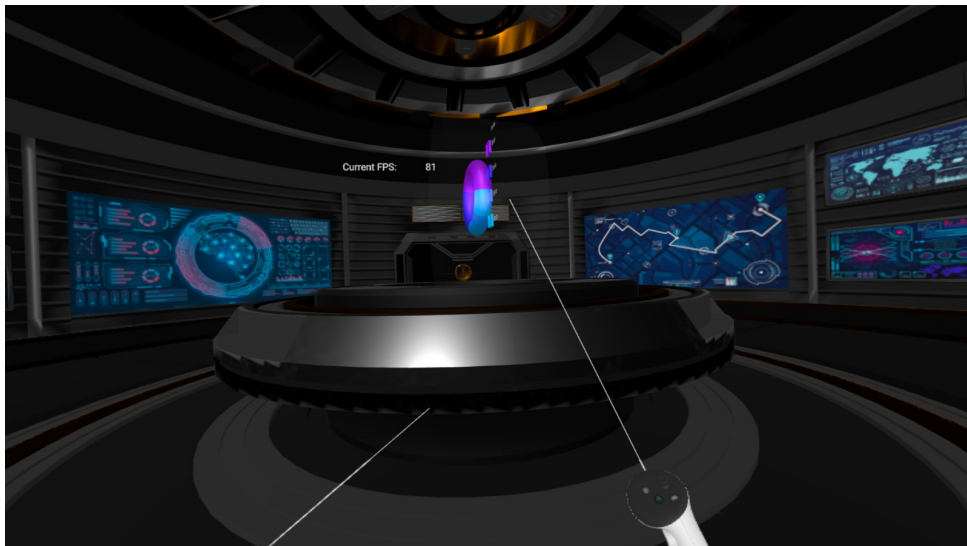


Figura 4.13: Interação de rotacionar ativa.
Fonte: Autoria própria

Interação de Redimensionar

As Figuras 4.14 e 4.15 ilustra um gráfico após sofrer alteração na sua escala.

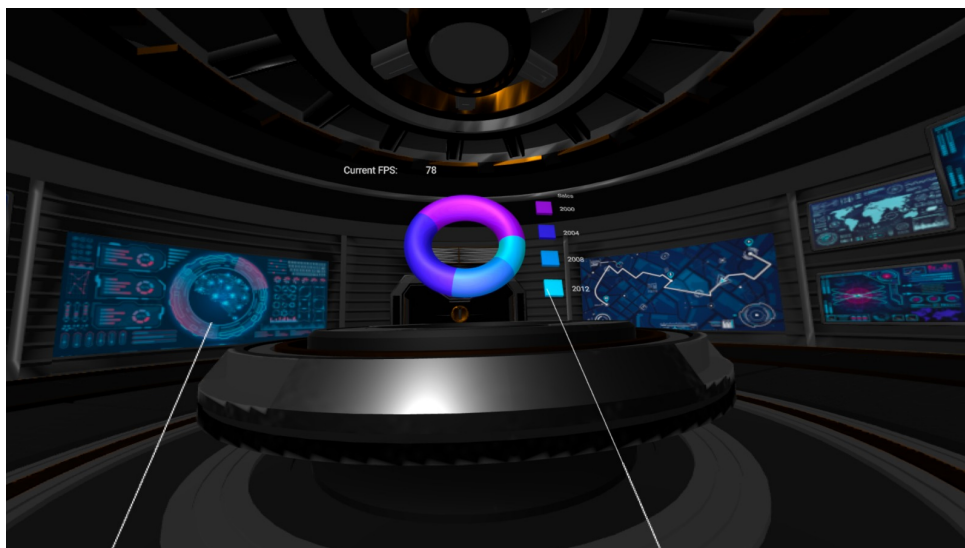


Figura 4.14: Interação de redimensionar aumentando.
Fonte: Autoria própria

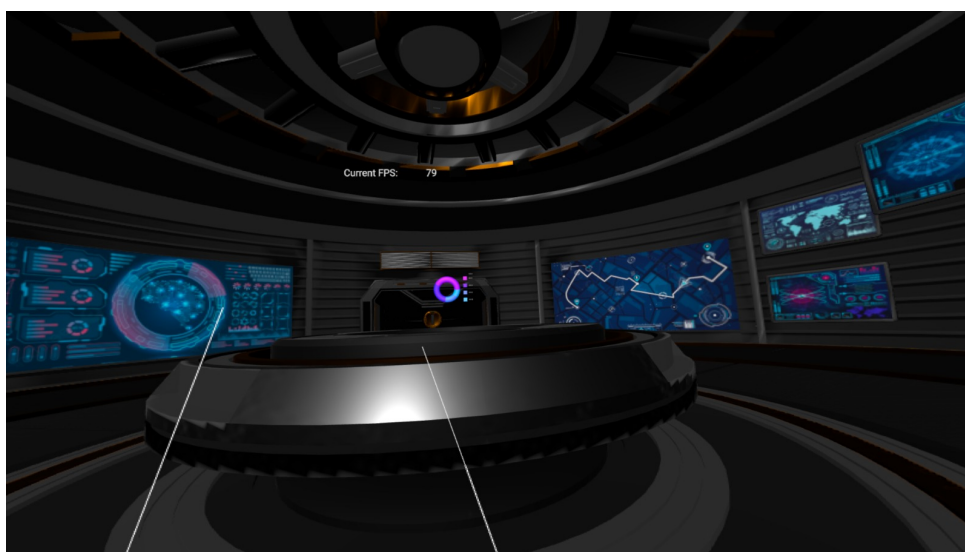


Figura 4.15: Interação de redimensionar diminuindo.
Fonte: Autoria própria

Interação de Filtrar

A Figura 4.16 ilustra um gráfico após ter dados filtrados.

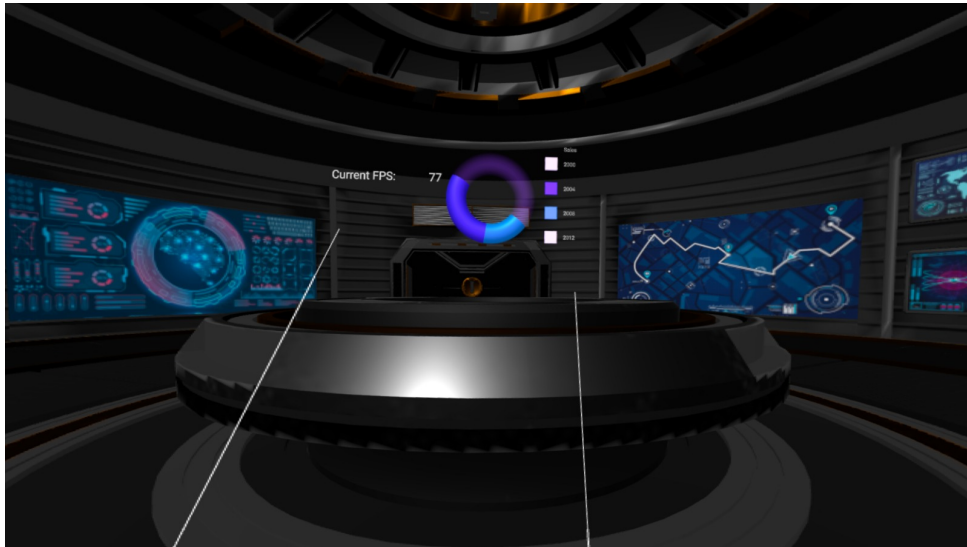


Figura 4.16: Interação de filtro ativa.
Fonte: Autoria própria

Efeito neon

A Figura 4.17 apresenta uma comparação entre os gráficos com e sem o efeito aplicado, ressaltando a diferença no impacto visual.

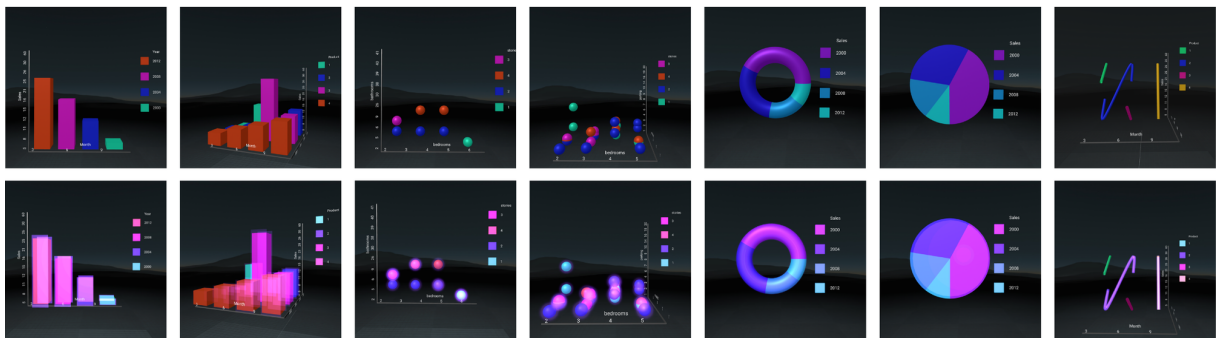


Figura 4.17: Antes e depois dos gráficos com o efeito de fake neon
Fonte: Autoria própria

Animações

A Figura 4.18 apresenta instantes da animação, exibindo o efeito gradual de escala.



Figura 4.18: Screenshots de instantes da animação
Fonte: Autoria própria

Modo AR

A Figura 4.19 apresenta o modo AR em funcionamento no Meta Quest 3.



Figura 4.19: Modo AR no Meta Quest 3
Fonte: Autoria própria

4.2.2 Demonstrações em vídeo

Para complementar as imagens, foram criados vídeos disponíveis no YouTube, que demonstram as principais funcionalidades da aplicação:

- **Utilizando a aplicação no modo VR:** Um vídeo detalhando a usabilidade de toda a aplicação no modo de realidade virtual foi gravado usando o Meta Quest 2. Acesso disponível em: <https://youtu.be/Vg7KsnaR7eA>
- **Utilizando a aplicação no modo AR:** Um vídeo detalhando a usabilidade de toda a aplicação no modo de realidade aumentada foi gravado usando o Meta Quest 3. Acesso disponível em: https://youtu.be/_gMnn-aIG3s

4.3 Testes em diferentes dispositivos

A fim de avaliar a aplicação desenvolvida, foram feitos testes em diferentes dispositivos: Meta Quest 2, Meta Quest 3, Smartphone e Notebook. A aplicação foi hospedada no *GitHub Pages* e acessada por meio de um *link* público. Nessa seção, serão apresentados os resultados obtidos, incluindo FPS médio, funcionalidades operantes, e modos disponíveis (AR ou VR). Imagens ilustrativas de cada teste também serão incluídas para melhor contextualização.

O Meta Quest 2 proporcionou uma experiência fluida tanto no modo VR (Figura 4.20) quanto no AR (Figura 4.21). No entanto, o AR apresentou limitações devido à câmera monocromática do dispositivo, que reduz a qualidade da imersão ao exibir imagens em preto e branco.

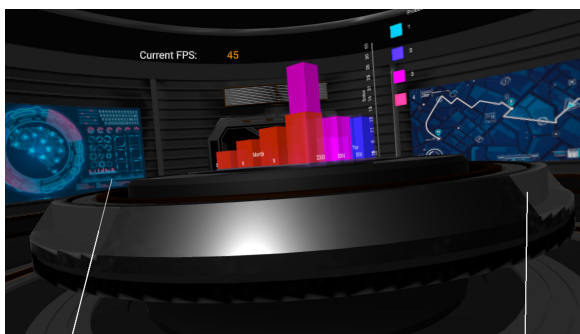


Figura 4.20: Aplicação no Meta Quest 2 no modo VR.

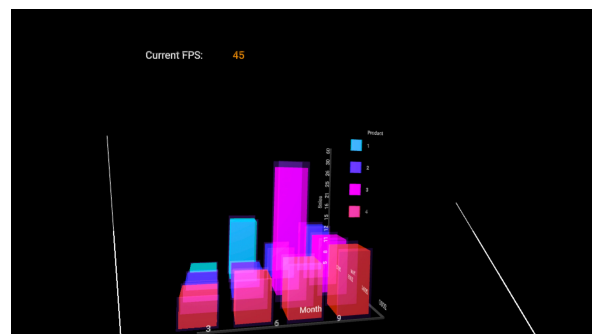


Figura 4.21: Aplicação no Meta Quest 2 no modo AR.

O Meta Quest 3 apresentou um desempenho superior, com FPS mais alto e excelente qualidade no modo AR, graças às câmeras coloridas avançadas. A transição entre os modos AR (Figura 4.23) e VR (Figura 4.22) foi suave e responsiva.

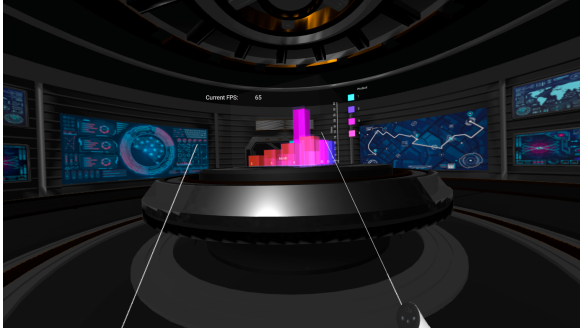


Figura 4.22: Aplicação no Meta Quest 3 no modo VR.

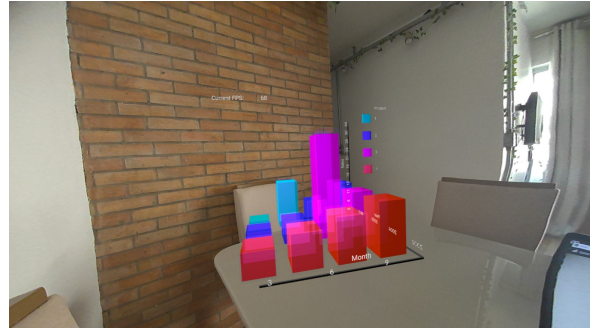


Figura 4.23: Aplicação no Meta Quest 3 no modo AR.

No smartphone (Figura 4.24) e no notebook (Figura 4.25), a aplicação foi limitada à visualização, sem suporte para interações. Apesar disso, o gráfico da tela inicial foi renderizado corretamente. No caso do último, no entanto, é possível interagir com o uso de simuladores de dispositivos VR no navegador.

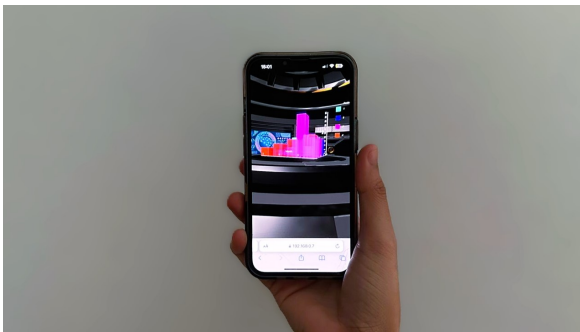


Figura 4.24: Tela inicial da aplicação em um smartphone.

Fonte: Autoria própria



Figura 4.25: Tela inicial da aplicação em um notebook.

Fonte: Autoria própria

Os resultados na tabela 4.2 indicam que a aplicação foi funcional em todos os dispositivos testados, com variações de desempenho e limitações específicas de cada *hardware*.

Dispositivo	FPS Médio	Funcionalidades Operantes	Modos Disponíveis
Meta Quest 2	45	Todas	AR monocromático, VR
Meta Quest 3	50	Todas	AR, VR
Notebook	150	Apenas visualização	3D, VR com simuladores
Smartphone	60	Apenas visualização	3D

Tabela 4.2: Resultados dos testes em diferentes dispositivos.

Capítulo 5

Conclusão

Neste trabalho, foram descritos os métodos, as técnicas e as ferramentas utilizadas no desenvolvimento de uma aplicação web para visualização de dados em realidade estendida, bem como os resultados alcançados e as dificuldades enfrentadas ao longo do caminho. O projeto desenvolvido teve como objetivo utilizar tecnologias emergentes, como XR, para criar um ambiente imersivo e interativo, visando aumentar a capacidade humana de reconhecer padrões e melhorar a percepção e o entendimento dos dados e, com isso, contribuir com a área de ciência de dados, especialmente no desafio de transformar dados em informação.

Dessa forma, a aplicação que funciona tanto em realidade virtual quanto em realidade aumentada foi desenvolvida utilizando o framework A-frame, que, por ser uma tecnologia de WebXR, é compatível com qualquer dispositivo com navegador e traz inúmeras vantagens. No total, cinco tipos diferentes de gráficos foram implementados: gráfico de barras, de dispersão, de linha, de donuts e de pizza. E três níveis de dimensionalidade foram explorados: 1D, 3D e 4D, com eixos e legendas complementares. Além disso, foram desenvolvidas quatro maneiras distintas de interagir com os gráficos, sendo elas: mover, rotacionar, redimensionar e filtrar. Também foi criada uma interface básica entre o usuário e a aplicação, com funcionalidades como um menu radial com botões para selecionar entre as opções disponíveis, HUDs informativas e feedbacks visuais que exibem o estado das interações. Por fim, a aplicação também dispõe de recursos adicionais, tais como a possibilidade de mudar de ambiente 3D e de eliminar as alterações feitas para retornar ao estado original.

Todo o sistema foi implementado seguindo padrões de arquitetura de software e recomendações de boas práticas da comunidade, como o uso de ECS. Assim, a aplicação resultante possui um código modular que possibilita a criação simplificada de novas visualizações de dados por meio de uma única linha de código com atributos personalizáveis. Toda a lógica que controla as interações, animações e exibições fica contida em seus componentes correspondentes. Isso torna o projeto mais escalável, facilitando a evolução do sistema, a manutenção e possibilitando a criação de novos tipos de gráficos, interações e

funcionalidades de maneira eficiente.

Foram feitos testes nos dispositivos Meta Quest 2, Meta Quest 3, em um smartphone e em um notebook. O resultado dos testes foram satisfatórios, apresentando FPS suficiente e interações funcionais. No entanto, as interações só funcionam em dispositivos de realidade estendida, com o uso de controles apropriados. Em dispositivos normais, só é possível visualizar um gráfico pré-exibido e andar pelo ambiente, sem interagir com os gráficos ou com a interface.

O trabalho desenvolvido permitiu explorar o uso da realidade estendida na visualização de dados, por meio de técnicas para exibição dos dados de maneira imersiva e da interação com esses dados, gráficos e ambientes. Sendo assim, espera-se que o trabalho desenvolvido apresente uma contribuição relevante para as áreas emergentes e correlacionadas, especialmente para pesquisadores das áreas de realidade estendida, que podem extrair conhecimento do desenvolvimento e de todo o processo seguido, e de visualização de dados, que tem a possibilidade de aumentar a eficácia do seu trabalho através de técnicas inovadoras.

Toda a pesquisa feita teve extrema importância para o conhecimento técnico da aluna, que conseguiu aplicar os conhecimentos adquiridos durante a sua graduação, como computação gráfica, álgebra linear, interação homem-máquina e desenvolvimento, arquitetura e engenharia de software. Além de aprender mais a fundo sobre as ferramentas com que trabalha e ter o contato com novas técnicas e tecnologias.

Para contribuições futuras e continuação do trabalho, as possibilidades são inúmeras. Podemos dividir em três principais categorias de *features*: visualização dos dados, interação com os dados e funcionalidade do sistema. Na primeira delas, uma grande prioridade seria desenvolver uma maneira de aceitar arquivos de dados diretamente pela aplicação e exibi-los como gráficos, pois dessa forma não seria necessário utilizar código para gerar os dados, aumentando o público-alvo que teria condições de utilizar a aplicação. Em seguida, desenvolver novos tipos de gráficos e ter uma biblioteca vasta. Também seria interessante explorar novas formas de representar mais dimensões, expandindo para além da quarta dimensão, através de atributos como textura, opacidade, tamanho, animação, cor do brilho, sons, feedbacks hápticos, etc.

Em relação à interação com os dados, há as seguintes possibilidades: Desenvolvimento de uma interação responsável por exibir informações extras ao clicar em um dado. Um comportamento de filtro análogo ao existente, onde o clique seria feito no próprio dado, para dar enfoque nele, e retirar os demais. Interação usando reconhecimento das mãos (*handtracking*) e explorando opções de gestos naturais, para uma interação cada vez mais natural e fluida. Mesclagem de gráficos, através do *merge* entre atributos. Adição de física nos dados e nos gráficos.

Quanto à terceira categoria, algumas ideias são: Adicionar um modo que permita a visualização e manipulação de mais de um gráfico por vez, a fim de permitir que o

usuário possa comparar informações. Melhorar a interface do usuário com a aplicação, adicionando mais feedbacks e huds informativas, além da possibilidade de adicionar voz e suporte para outros idiomas. Adicionar um *canvas* ou *whiteboard*, para permitir que o usuário faça marcações ou anotações sem precisar sair da imersão. Adicionar possibilidade de *multi-players* em uma mesma seção, a fim de aumentar a colaboração e aumentar o alcance da aplicação, tendo em vista que apenas uma pessoa precisaria estar em um dispositivo imersivo e o restante poderia acompanhar por um dispositivo móvel. Adição de *navmesh* para restringir os espaços livres na sala e adicionar colisões, sempre pensando em aumentar o realismo. Por fim, suporte para dispositivos não imersivos, permitindo a interação e a visualização dos dados através da aplicação e de interação sem controles.

Bibliografia

- 3DDisco (2024). Sci-fi command room interior. <https://sketchfab.com/3d-models/sci-fi-command-room-interior-3d-model-22c3ec5244aa4f52b7c7ef74061fe4ca>. Modelo 3D disponibilizado no Sketchfab. Acessado em: 18 nov. 2024.
- Chen, J. X. (2005). Data visualization and virtual reality. *Handbook of Statistics*, 24:539–563. Artigo Historico, conjunto de ferramentas, olhar a estrutura.
- Dibbern, C., Uhr, M., Krupke, D., and Steinicke, F. (2018). Can webvr further the adoption of virtual reality? *Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI)*, pages 377–384.
- DiBenigno, M., Kosa, M., and Johnson-Glenberg, M. C. (2021). Flow immersive: A multiuser, multidimensional, multiplatform interactive covid-19 data visualization tool. *Frontiers in Psychology*, 12:661613.
- Donalek, C., Djorgovski, S. G., Cioc, A., Wang, A., Zhang, J., Lawler, E., Yeh, S., Mahabal, A., Graham, M., Drake, A., Davidoff, S., Norris, J. S., and Longo, G. (2014). Immersive and collaborative data visualization using virtual reality platforms. *IEEE Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014*, pages 609–614.
- ektogamat (2024). Fake glow material for react three fiber. <https://github.com/ektogamat/fake-glow-material-r3f>. Acessado em: 18 nov. 2024.
- Hadjar, H., Meziane, A., Gherbi, R., Setitra, I., and Aouaa, N. (2018). Webvr based interactive visualization of open health data. In *Proceedings of the 2nd International Conference on Web Studies*, WS.2 2018, page 56–63, New York, NY, USA. Association for Computing Machinery.
- Lee, B., Bach, B., Dwyer, T., and Marriott, K. (2019). Immersive analytics. *IEEE Computer Graphics and Applications*, 39:16–18.
- Liu, K., Wu, N., and Han, B. (2023). Demystifying web-based mobile extended reality accelerated by webassembly. In *Proceedings of the 2023 ACM on Internet Measurement*

- Conference*, IMC '23, page 145–153, New York, NY, USA. Association for Computing Machinery.
- Liu, R., Gao, M., Wang, L., Wang, X., Xiang, Y., Zhang, A., Xia, J., Chen, Y., and Chen, S. (2022). Interactive extended reality techniques in information visualization. *IEEE Transactions on Human-Machine Systems*, 52(6):1338–1351.
- Moran, A., Gadepally, V., Hubbell, M., and Kepner, J. (2015). Improving big data visual analytics with interactive virtual reality. *2015 IEEE High Performance Extreme Computing Conference, HPEC 2015*.
- Moreno-Lumbreras, D., Gonzalez-Barahona, J. M., and Robles, G. (2023). Babiaxr: Facilitating experiments about xr data visualization. *SoftwareX*, 24:101587.
- Moreno-Lumbreras, D., Gonzalez-Barahona, J. M., and Villaverde, A. (2022). Babiaxr: Virtual reality software data visualizations for the web. *Proceedings - 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, VRW 2022*, pages 71–74.
- Olshannikova, E., Ometov, A., Koucheryavy, Y., and Olsson, T. (2015). Visualizing big data with augmented and virtual reality: challenges and research agenda. *Journal of Big Data*, 2:1–27.
- Shakeel, H. M., Iram, S., Al-Aqrabi, H., Alsboui, T., and Hill, R. (2022). A comprehensive state-of-the-art survey on data visualization tools: Research developments, challenges and future domain specific visualization framework. *IEEE Access*, 10:96581–96601.
- Tamayo, J. L. R., Hernández, M. B., and Gómez, H. G. (2018). Digital data visualization with interactive and virtual reality tools. review of current state of the art and proposal of a model. *Revista ICONO 14. Revista científica de Comunicación y Tecnologías emergentes*, 16:40–65.
- Vasarainen, M., Paavola, S., and Vetoshkina, L. (2021). A systematic literature review on extended reality: Virtual, augmented and mixed reality in working life. *International Journal of Virtual Reality*, 21(2):1–28.
- Wahidi, M., Ahmad, T., and Litchmore, T. (2023). Big data visualization in extended reality.