

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

VICTORIA DE AVELAR DUARTE

Aplicação Web para Acervo do INF

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em Ciência
da Computação

Orientador: Prof. Dr. Marcelo Soares Pimenta

Porto Alegre
2024

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões Mendes

Vice-Reitora: Prof^ª. Patricia Helena Lucas Pranke

Pró-Reitora de Graduação: Prof^ª. Cíntia Inês Boll

Diretora do Instituto de Informática: Prof^ª. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Marcelo Walter

Bibliotecário-chefe do Instituto de Informática: Alexsander Borges Ribeiro

RESUMO

Este trabalho descreve a criação da plataforma "Memórias em Bits" para o Instituto de Informática da Universidade Federal do Rio Grande do Sul. O projeto tem dois objetivos principais: em primeiro lugar, a construção de um acervo digital abrangente, que cataloga os itens físicos presentes no Instituto; e, em segundo lugar, a utilização de estratégias de gamificação para enriquecer a experiência dos visitantes ao acervo (digital ou físico), orientando-os por exposições e incentivando a exploração interativa do Instituto de Informática. O presente volume descreve em detalhes o processo de definição e construção desse acervo com gamificação, assim como conceitos, ferramentas e tecnologias que foram adotados no decorrer do trabalho.

Palavras-chave: Aplicação web. gamificação. acervo. INF UFRGS.

Web Application for INF Collection

ABSTRACT

This work describes the creation of the "Memórias em bits" platform for the Institute of Informatics at the Federal University of Rio Grande do Sul. The project has two main objectives: first, the construction of a comprehensive digital collection that catalogs the physical items present in the Institute; and second, the use of gamification strategies to enhance the visitor's experience with the collection (digital or physical), guiding them through exhibitions and encouraging interactive exploration of the Institute of Informatics. This volume details the process of defining and building this gamified collection, as well as the concepts, tools, and technologies adopted throughout the project.

Keywords: web application. gamification. exhibition. INF UFRGS.

LISTA DE FIGURAS

Figura 2.1 Museu de Ciência do Reino Unido: atração Power Up. O quiosque em foco, contém o jogo <i>Halo</i> , um jogo de tiro que pode ser jogado por até 16 jogadores.	14
Figura 2.2 Museu do Amanhã: atração Cosmos.	14
Figura 2.3 Museu de História Natural Americano: a utilização do aplicativo complementa a experiência presencial do usuário.	15
Figura 2.4 Museu Mariano Procópio: em algum dos objetos, foi adicionado um QR Code junto a placa de informações. Ao scanear o código, é possível obter informações extras sobre a peça em exposição.	16
Figura 3.1 Os quatro tipos de jogadores segundo Bartle.	19
Figura 4.1 Página principal do acervo digital, com o catálogo de itens e os filtros disponíveis.	25
Figura 4.2 Os filtros podem ser expandidos. As opções de marca e categoria disponíveis para seleção aparecem de acordo com as opções existentes no banco de dados.	26
Figura 4.3 Página de detalhes de um item.	26
Figura 4.4 Página geral de progresso do usuário. É possível ver a pontuação atual e seu progresso geral.	28
Figura 4.5 Página de progresso das conquistas.	28
Figura 4.6 Página de detalhes de uma conquista.	29
Figura 4.7 Tabela com as informações sobre itens, dentro da página de administração.	30
Figura 4.8 Formulário para adição de uma exibição.	30
Figura 5.1 Arquitetura do sistema.	32
Figura 5.2 Fluxo do registro do usuário no aplicativo.	33
Figura 5.3 Diagrama ER.	34
Figura 5.4 Estrutura de pastas da API do projeto.	37
Figura 5.5 Estrutura de pastas do <i>frontend</i> do projeto.	38
Figura 5.6 Organização da pasta <i>view</i>	40
Figura 5.7 Organização típica de uma componente.	40
Figura 5.8 Estrutura de páginas do <i>frontend</i>	41
Figura 5.9 Fluxo da experiência do usuário no aspecto de gamificação.	42
Figura 5.10 Cartas de troca do jogo <i>Chicory: a colorful tale</i>	44
Figura 6.1 Informações demográficas sobre os entrevistados.	49
Figura 6.2 Avaliação da dificuldade dos usuário ao realizar a tarefa 1, sendo 1 muita dificuldade e 5 muita facilidade.	49
Figura 6.3 Avaliação da dificuldade dos usuário ao realizar a tarefa 2, sendo 1 muita dificuldade e 5 muita facilidade.	50
Figura 6.4 Avaliação da dificuldade dos usuário ao realizar a tarefa 3, sendo 1 muita dificuldade e 5 muita facilidade.	50
Figura 6.5 Avaliação da dificuldade dos usuário ao realizar a tarefa 4, sendo 1 muita dificuldade e 5 muita facilidade.	51
Figura 6.6 Avaliação da dificuldade dos usuário ao realizar a tarefa 5, sendo 1 muita dificuldade e 5 muita facilidade.	51
Figura 6.7 De modo geral, a aplicação foi bem recebida pelos usuários que a testaram.	52

Figura 7.1 A maioria dos usuários declarou que usaria a aplicação53

LISTA DE TABELAS

Tabela 5.1	Tabela dos atributos base esperados para o modelo de conquista	35
Tabela 5.2	Tabela dos atributos base esperados para o modelo de cartas	36
Tabela 5.3	Tabela dos atributos base esperados para o modelo de exibição	39
Tabela 5.4	Tabela dos atributos base esperados para o modelo de item	46
Tabela 5.5	Tabela dos atributos base esperados para o modelo de categoria de item	46
Tabela 5.6	Tabela dos atributos base esperados para o modelo de quiz.....	46
Tabela 5.7	Tabela dos atributos base esperados para o modelo de rotas.....	47
Tabela 5.8	Tabela dos atributos base esperados para o modelo de usuário.....	47

LISTA DE ABREVIATURAS E SIGLAS

- DBaaS *Database As A Service* (Base de Dados como Serviço)
- UFRGS Universidade Federal do Rio Grande do Sul
- CI/CD *Continuous Integration/ Continuous Delivery* (Integração Contínua/ Entrega Contínua)
- CRUD *Create Read Update Delete* (Criar Ler Atualizar Deletar)
- API *Application Programming Interface* (Interface de Programação de Aplicação)
- AWS *Amazon Web Services* (Serviços Web da Amazon)
- INF Instituto de Informática da UFRGS
- MVP *Minimum Viable Product* (Produto Viável Mínimo)
- PET Programa de Educação Tutorial

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Objetivo	11
1.2 Estrutura do texto	11
2 TRABALHOS RELACIONADOS	13
2.1 Museu de Ciência do Reino Unido	13
2.2 Museu do Amanhã	13
2.3 Museu de História Natural Americano	13
2.4 Museu Mariano Procópio	15
2.5 Conclusão	15
3 CONCEITOS E TECNOLOGIAS	17
3.1 MVP	17
3.2 Gamificação	17
3.2.1 Mecânicas de gamificação	17
3.2.2 Jogador	18
3.3 Aplicação <i>web</i>	19
3.4 API RESTful	19
3.5 Banco de Dado NoSQL	20
3.6 Tecnologias	20
3.6.1 Atlas	20
3.6.2 AWS	20
3.6.3 Axios	21
3.6.4 CI/CD	21
3.6.5 CRUD	21
3.6.6 MongoDB	21
3.6.7 Mongoose	22
3.6.8 Node.js	22
3.6.9 OAuth 2.0	22
3.6.10 ReactJS	22
3.6.11 React Router	23
3.6.12 Styled Components	23
3.6.13 Typescript	23
4 MUSEU VIRTUAL: VISÃO GERAL	24
4.1 Requisitos	24
4.2 Fluxo geral do projeto	25
4.2.1 Acervo digital	25
4.2.2 Exploração das exposições	27
4.2.2.1 Escolha da rota	27
4.2.2.2 Desbloqueio de exibição	27
4.2.2.3 Acompanhamento do próprio progresso	27
4.2.3 Administração	29
5 MUSEU VIRTUAL: DESENVOLVIMENTO E IMPLEMENTAÇÃO	31
5.1 Arquitetura	31
5.1.1 Fluxo de cadastro de usuário	31
5.2 Banco de Dados	32
5.2.0.1 Aquisição dos Dados	33
5.2.1 Construção dos modelos	33
5.2.1.1 <i>Achievements</i>	33
5.2.1.2 <i>Cards</i>	34

5.2.1.3	<i>Exhibition</i>	34
5.2.1.4	<i>Items</i>	34
5.2.1.5	<i>Categoria de item</i>	34
5.2.1.6	<i>Quizes</i>	35
5.2.1.7	<i>Routes</i>	35
5.2.1.8	<i>User</i>	35
5.3	API	36
5.4	Frontend	36
5.5	Gamificação	41
5.5.1	Pontos	41
5.5.2	Conquistas	43
5.5.3	Encontrar exposições físicas	43
5.5.4	Coleções de cartas	43
5.5.5	Roteiro de visitas	43
5.6	Hospedagem	44
5.6.1	Banco de Dados	44
5.6.2	Backend	44
5.6.3	Frontend	45
6	AVALIÇÃO	48
7	CONCLUSÃO	53
7.1	Resultados obtidos	53
7.2	Limitações	54
7.2.1	Login	54
7.2.2	Funcionalidades não implementadas	54
7.2.2.1	Eventos	54
7.2.2.2	Interação entre usuários	54
7.2.3	Experiência de usuário e navegação	54
7.2.4	Amostragem	55
7.3	Trabalhos futuros	55
	REFERÊNCIAS	56

1 INTRODUÇÃO

No prédio da administração, o Instituto de Informática guarda um grande tesouro. Uma pequena sala, com o vidro da porta coberto por papel preto, está lotada com diversos itens históricos do universo da tecnologia. Esses objetos, coletados ao longo dos anos por professores e alunos, variam de cartões perfurados, Macintosh, os primeiros celulares Motorolas e até Macbooks da última década.

A ideia de criar um museu no Instituto de Informática é antiga. Mais recentemente, em 2022, a direção do INF entrou em contato com o grupo PET Computação UFRGS ¹ com a iniciativa de expor os itens do acervo. Inicialmente, foram selecionados alguns consoles de jogos para serem expostos durante o Portas Abertas. Com esse teste inicial bem sucedido, o próximo passo seria classificar outros itens e montar exposições permanentes e itinerantes.

1.1 Objetivo

O projeto apresentado neste texto é um complemento do trabalho iniciado pelo PET e pela Direção do Instituto de Informática. Os objetivos são desenvolver um aplicativo de acervo digital, oferecer um complemento de informações que auxiliam a visita ao museu digital e utilizar gamificação para enriquecer a experiência dos visitantes ao explorarem esse acervo, seja de forma física ou virtual.

1.2 Estrutura do texto

O restante deste texto está organizado da seguinte maneira: o Capítulo 2 cita e explora outros trabalhos relacionados, comparando-os ao presente trabalho; o Capítulo 3 aborda conceitos, ferramentas e tecnologias que foram utilizadas para o desenvolvimento deste projeto; o Capítulo 4 descreve em mais detalhes o processo de construção dessa ferramenta, a motivação para sua construção e o levantamento de requisitos; o Capítulo 5

¹O Programa de Educação Tutorial (PET) é um programa do Governo Federal brasileiro que visa estimular as atividades de pesquisa, ensino e extensão dentro das universidades em nível de graduação.

O PET Computação da UFRGS é composto por alunos de graduação dos cursos de Ciência e Engenharia da Computação e tem como principal meta promover oportunidades para seus membros explorarem suas áreas de interesse além do que já é oferecido no currículo, retornando o conhecimentos adquirido à comunidade. (PET COMPUTAÇÃO UFRGS,)

descreve o processo de desenvolvimento e o funcionamento da versão final; o Capítulo 6 avalia os resultados obtidos com usuários reais e; por fim, o Capítulo 7 revisa as conclusões e contribuições deste trabalho e apresenta sugestões de possíveis trabalhos futuros.

2 TRABALHOS RELACIONADOS

Museus interativos vêm ganhando cada vez mais espaço no século XXI. O termo museu interativo pode ser usado tanto para instituições digitais, encontradas online, quanto para físicas, que utilizam as tecnologias digitais para ampliar a experiência presencial. Há diversos museus interativos espalhados pelo Brasil e mundo, que incorporam essa interatividade de diferentes formas e neste capítulo apresentaremos resumidamente alguns deles.

2.1 Museu de Ciência do Reino Unido

O Museu de Ciência do Reino Unido, por exemplo, tem a exposição permanente *Power Up*, que conta com 160 consoles com os melhores jogos dos últimos 50 anos, todos acessíveis aos visitantes. Os jogos vão de Pong e Pacman até Minecraft e Mario Kart. Os jogos são divididos em diversas categorias, ou pela quantidade de jogadores que podem jogar em uma mesma partida, como mostrado na figura 2.1. (SCIENCE MUSEUM UK,)

2.2 Museu do Amanhã

Ainda pensando na imersão do visitante, o Museu do Amanhã, localizado no Rio de Janeiro, é um museu de ciência focado nos desafios que a humanidade terá que enfrentar no futuro com relação à sustentabilidade. Em uma de suas atrações principais, Cosmos, os visitantes entram em um domo com uma projeção em 360 graus, tal qual mostrado na imagem 2.2. Em oito minutos, é contada desde a formação da Terra até o desenvolvimento da vida e pensamento pela manifestação pela arte. (MUSEUS, 2020)

2.3 Museu de História Natural Americano

A interatividade não precisa ser apenas a imersão do visitante. O Museu de História Natural Americano, por exemplo, conta com um aplicativo que pode ser usado dentro dos prédios. Nele, é possível ver a localização de exposições e outros pontos importantes, como banheiros e lojas. Além disso, é possível receber recomendações personalizadas sobre quais atrações visitar e, assim como mostrado na figura 2.3, mais informações so-

Figura 2.1: Museu de Ciência do Reino Unido: atração Power Up. O quiosque em foco, contém o jogo *Halo*, um jogo de tiro que pode ser jogado por até 16 jogadores.



Fonte: Science Museum Group

Figura 2.2: Museu do Amanhã: atração Cosmos.



Fonte: Museu do Amanhã

Figura 2.3: Museu de História Natural Americano: a utilização do aplicativo complementa a experiência presencial do usuário.



Fonte: Emon Hassan para The New York Times

bre os objetos presentes nas exposições físicas. (AMERICAN MUSEUM OF NATURAL HISTORY,)

2.4 Museu Mariano Procópio

O Museu Mariano Procópio, em Juiz de Fora, por sua vez, incluiu a interatividade de maneira simples. A figura 2.4 mostra como a inclusão foi feita, através do uso de QR Codes em alguns dos objetos expostos. O objetivo é que o visitante possa, ao escanear o código, ter acesso a conteúdos extras, complementando as informações presentes fisicamente. (MATA; MGTV, 2018)

2.5 Conclusão

Há várias formas de adicionar interatividade à exposições e cada museu escolheu diferentes formas de incorporar essa interatividade. Os museus apresentados foram uma fonte de inspiração para construir a interatividade desenvolvida no trabalho.

Embora não seja usado diretamente jogos, como no Museu de Ciência do Reino Unido, a exposição *Power Up* motivou o uso de gamificação para o desenvolvimento do

Figura 2.4: Museu Mariano Procópio: em algum dos objetos, foi adicionado um QR Code junto a placa de informações. Ao scanear o código, é possível obter informações extras sobre a peça em exposição.



Fonte: Prefeitura de Juiz de Fora

projeto.

Além disso, assim como o Museu de História Natural, foi pensado em uma aplicação para ser utilizada durante a visita. Por fim, também foi adotado o uso de QR Codes, como foi feito no Museu Mariano Procópio, permitindo aos visitantes acessarem páginas online que complementam as informações do ambiente físico.

3 CONCEITOS E TECNOLOGIAS

Este capítulo visa fornecer uma compreensão dos conceitos necessários para a compreensão deste trabalho.

Uma vez que o produto final deste trabalho é composto por três aplicações - o *frontend*, a API e o banco de dados - e pela adoção de gamificação, estes conceitos serão primeiramente explicados e algumas das tecnologias envolvidas serão resumidas.

3.1 MVP

Um MVP é um produto de baixo custo e desenvolvimento em um curto período de tempo. É uma versão reduzida do produto final focado em validar um modelo de negócio. O objetivo é criar um produto já utilizável pelo cliente desde o primeiro lançamento, mesmo se for necessário várias rodadas de *feedback*.

A agilidade na entrega permite validar com o cliente as funcionalidades que são adicionadas de maneira imediata. Também é possível alinhar expectativas e prioridades conforme a aplicação é desenvolvida, garantindo mais flexibilidade em alterações e mudanças de prioridade. (SANTOS, 2018)

3.2 Gamificação

A gamificação se trata de usar mecânicas de jogos, como pontos, conquistas, níveis, placares e coleções, fora do ambiente de um jogo. O objetivo dessas recompensas é engajar e motivar pessoas a atingir um determinado objetivo, tornando uma experiência comum mais motivadora. (BURKE, 2015)

3.2.1 Mecânicas de gamificação

No âmbito da gamificação, a diversão não depende do tema da aplicação, mas nas mecânicas incorporadas. Compreender a motivação dos jogadores é fundamental para o desenvolvimento de mecânicas atraentes e envolventes. Acredita-se que existem quatro motivos principais que impulsionam as pessoas a se envolverem em jogos: a busca pelo aprimoramento, o alívio do estresse, a busca pela diversão e a oportunidade de socializar.

Há várias mecânicas disponíveis que poderiam ser adotadas para o desenvolvimento da aplicação. Por se tratar de um MVP, foi necessário selecionar apenas algumas:

- **Pontos:** representa numericamente o progresso do usuário e podem ser obtidos através de ações realizadas por ele, como se registrar na aplicação e encontrar exposições. Nessa aplicação, os pontos são usados como moeda, sendo possível comprar cartas usando os pontos obtidos;
- **Conquistas:** são dadas aos usuários ao completarem tarefas específicas. Possibilitam o usuário, de forma visual, medir seu progresso e o recompensar por realizar determinadas ações. Algumas das conquistas que podem ser desbloqueadas são: ter se registrado no aplicativo, desbloquear uma exposição ou desbloquear um certo número de exposições;
- **Coleções:** cria um senso de posse com a coleta de itens. No projeto, a coleção corresponde a coleção de cartas de cada exposição;
- **Missões:** são objetivos pré-definidos que possuem recompensas. Missões podem ser conquistadas em um curto período de tempo e costumam fazer parte de um objetivo maior. No contexto da aplicação, cada exposição que pode ser encontrada corresponde a uma missão. O objetivo maior é desbloquear todas as exposições;

(KLOCKA et al., 2020)

3.2.2 Jogador

Além de compreender as motivações, é crucial determinar para qual tipo de jogador a aplicação será direcionada. O trabalho de Bartle (2003) categoriza os jogadores em quatro tipos distintos, representados na figura 3.1.

Os *explorers* (exploradores) têm uma inclinação para explorar o universo do jogo, buscando adquirir experiência. Os *achievers* (conquistadores) anseiam por conquistas e vitórias. *Socializers* (socializadores), como o tipo mais comum, valorizam a interação social com outros jogadores. Já os *killers* (matadores) compartilham o desejo de vitória com os *achievers*, mas buscam também a derrota de outros e o respeito de outros jogadores.

Figura 3.1: Os quatro tipos de jogadores segundo Bartle



Fonte: Interaction Design Foundation

3.3 Aplicação web

Uma aplicação *web* é um tipo de *software* que opera diretamente em navegadores *web*. Esse tipo de aplicação oferece diversas vantagens, como a compatibilidade com a maioria dos computadores e sistemas operacionais comuns, acessibilidade em diversos dispositivos, incluindo *desktop* e dispositivos móveis, baixa ocupação de espaço na memória do disco e a capacidade de atender a vários usuários simultaneamente. A principal desvantagem reside na dependência de uma conexão com a internet, embora esse desafio tenha diminuído ao longo dos anos devido ao aprimoramento das redes.

As aplicações *web* são tipicamente divididas em duas faces: o *frontend*, ou *client-side*, responsável por apresentar informações ao usuário, e o *backend*, ou *server-side*, encarregado do processamento de dados. (AWS, b)

3.4 API RESTful

Uma Interface de Programação de Aplicação (API) é uma interface usada por sistemas para trocar informações, definindo regras para que esses sistemas possam se co-

municar. O termo REST vem de Transferência de Estado Representacional, que é uma arquitetura de *software* que define como a API deve funcionar. As APIs RESTful promovem escalabilidade, flexibilidade e independência, otimizando as interações entre cliente e servidor.(AWS, a)

3.5 Banco de Dado NoSQL

Bancos de dados NoSQL são bancos que armazenam dados de forma mais flexível, sendo ideais para lidar com informações que não se enquadram em modelos específicos, como os utilizados em bancos de dados SQL. Oferece flexibilidade, escalabilidade e alto desempenho, trazendo grande facilidade no desenvolvimento, já que bancos NoSQL se adaptam facilmente ao desenvolvimento ágil e as mudanças frequentes dos requisitos. Além disso, são necessárias menos transformações no código, já que os dados são armazenados de maneira mais próxima a como são utilizados em aplicações.(ORACLE,)

3.6 Tecnologias

3.6.1 Atlas

É um serviço DBaaS para gerenciar a infraestrutura e segurança, além de fazer a manutenção das máquinas. Ele cuida da parte de configuração, *deploy* e escalabilidade do banco de dados em um serviço de nuvem. (MONGODB, a)

3.6.2 AWS

A AWS é uma plataforma de cloud computing da Amazon no formato *pay-per-use*, que oferece soluções em diversas áreas, como armazenamento de dados, infraestrutura de redes e hospedagem. Para esse trabalho, vale destacar o EC2 e o S3. O EC2 é o serviço de nuvem, que possibilita a criação e configuração de servidores virtuais. Já o S3 é o serviço de armazenamento que utiliza *buckets* para armazenar os dados.(BESSA, a)

3.6.3 Axios

Axios é um cliente HTTP baseado em promessas que não depende de frameworks ou bibliotecas. Super versátil, pode ser usado tanto no lado do cliente quanto do lado do servidor. No lado do servidor, são utilizadas requisições HTTP, enquanto, no lado cliente, são usadas requisições XMLHttpRequests(AXIOS,)

3.6.4 CI/CD

CI/CD é uma prática de desenvolvimento de software que tem como objetivo tornar a integração de código mais simples com *builds* e testes automatizados. O CI/CD monitora todo o ciclo de vida da aplicação, adicionando etapas de teste, integração e automação.(REDHAT, 2023)

3.6.5 CRUD

CRUD é um mnemônico para as operações básicas usadas no armazenamento persistente, normalmente realizadas em banco de dados ou aplicações web. É com essas operações que o usuário pode interagir com o sistema.(MOZILLA, 2023)

3.6.6 MongoDB

O MongoDB é um sistema de gerenciamento de banco de dados NoSQL, orientado a documentos. Seus dados são armazenados em documentos flexíveis do tipo BSON (Binary JSON), estruturados em pares de chave-valor e agrupados em coleções, trazendo um formato semelhante ao JSON. Sua flexibilidade permite que documentos em uma coleção possam ter diferentes campos, sem a necessidade de uma estrutura rígida predefinida.(MONGODB, b)

3.6.7 Mongoose

Mongoose é uma biblioteca de Modelagem de Dados de Objeto para MongoDB no Node.js. A biblioteca proporciona uma camada de abstração que simplifica a interação com o banco de dados, gerenciando relacionamentos, validando esquemas e traduzindo objetos entre código e MongoDB.(ESPPENCHUTZ, 2022)

3.6.8 Node.js

É um ambiente de execução Javascript que permite a criação de aplicações não bloqueadoras orientadas a eventos. Também possibilita transferir para o lado do servidor o processamento e renderização do código. Isso permite a criação de aplicações em uma máquina servidora, sem a necessidade de um navegador.(BESSA, b)

3.6.9 OAuth 2.0

OAuth 2.0 é um protocolo de autorização que permite uma aplicação acessar recursos hospedados em outros aplicativos *web*, em nome de um usuário. É muito utilizado para fazer *login* em aplicações e na autenticação de APIs, utilizando tokens de acesso. Esses tokens representam a autorização para acessar recursos específicos em nome do usuário. Os tokens não tem um formato definido, mas é *JSON Web Token (JWT)* é o formato frequentemente usado.(OKTA,)

3.6.10 ReactJS

O React, uma biblioteca de JavaScript, tem como objetivo simplificar a criação de interfaces de usuário (UI), ao fundir de maneira mais acessível o HTML e o JavaScript. Esta simplificação é alcançada por meio da utilização de componentes, fragmentos de código isolados e reutilizáveis.

Quando uma página é carregada, o navegador cria uma representação conhecida como DOM (Document Object Model) do HTML, organizando os elementos como objetos em uma árvore. Contudo, o DOM apresenta a desvantagem de requerer atualizações frequentes toda vez que qualquer alteração é feita em algum elemento do documento

HTML, o que pode resultar em um custo significativo no desempenho.

O React otimiza esse processo de atualização do DOM ao introduzir o conceito de um Virtual DOM. Através da implementação de um Virtual DOM, é possível reduzir as atualizações diretas no DOM, porque o Virtual DOM reflete apenas as modificações que necessitam ser incorporadas no DOM, evitando atualizações desnecessárias e, assim, melhorando eficiência do sistema. (LONGEN, 2023)

3.6.11 React Router

O ReactJS não oferece uma ferramenta nativa para navegação entre as páginas da aplicação. Por esse motivo, usa-se o React Router, uma biblioteca para gerenciamento de rotas, que permite a criação de caminhos, renderização de componentes e resolução de parâmetros na URL.(SOUSA, 2022)

3.6.12 Styled Components

Styled Components é uma biblioteca que permite ao desenvolvedor escrever CSS dentro do Javascript, especialmente usada com React e React Native. A biblioteca possibilita injetar apenas o CSS que será utilizado, evitando carregar estilos desnecessários. Além disso, com os nomes de classes são gerados automaticamente, evitando nomes de classes duplicadas.(TV, 2022)

3.6.13 Typescript

TypeScript é um conjunto de ferramentas para o JavaScript, conhecida como um *superset*. Utilizado no desenvolvimento e convertido totalmente para JavaScript, tem como objetivo tornar o desenvolvimento mais seguro e facilitar a manutenção do código, incorporando recursos como tipagem estática, suporte para interfaces, classes, herança e módulos.(ROVEDA, 2023)

4 MUSEU VIRTUAL: VISÃO GERAL

O contato inicial com o que se tornaria o trabalho apresentado nesse texto se deu com o PET Computação. O PET possuía um projeto *Acervo*, em parceria com a Direção do Instituto de Informática, que tinha como objetivo fazer a catalogação dos itens presentes da sala do acervo do INF. Além da catalogação, seriam criadas exposições físicas pelo Instituto. Pensou-se em uma expansão para o projeto existente, que se tornaria o trabalho apresentado no texto. Nas conversas, então, foram definidos objetivos e requisitos para o projeto.

4.1 Requisitos

A plataforma construída tem como objetivos: primeiro, aproveitar a catalogação que já seria feita e desenvolver um acervo digital que pudesse ser acessado pelo público; segundo, incentivar a exploração das exposições espalhadas pelos prédios usando gamificação.

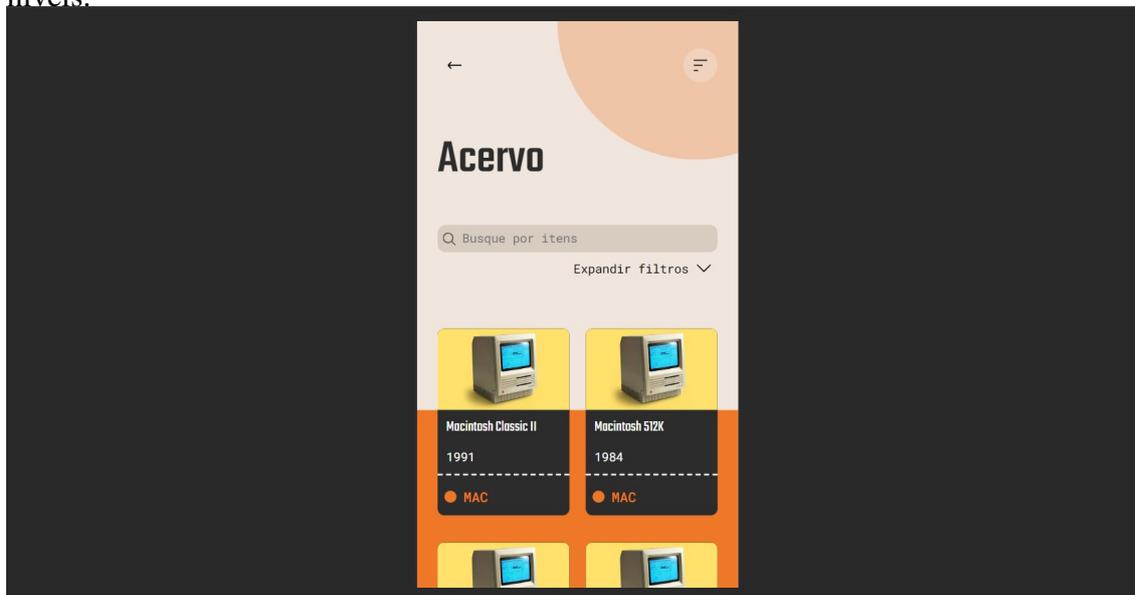
Para a realização do produto final, a primeira etapa foi definir um público-alvo. O Instituto de Informática é formado por um conjunto de prédios um pouco mais afastado dos outros prédios do campus Vale, que, por sua vez, é um campus localizado no bairro Agronomia, mais distante do centro de Porto Alegre.

Essa distância, faz com que a maioria dos frequentadores das instalações sejam os próprios alunos da computação. Porém, a UFRGS, no geral, é responsável por promover diversos eventos abertos ao público dentro do ambiente universitário, como o Portas Abertas, por exemplo. Dessa forma, é fundamental garantir que a solução criada também dialogue com um público mais geral.

Assim, pensando que a solução deve atender a um público geral, houveram duas grandes preocupações. A primeira é que a aplicação seja simples e intuitiva de usar, então a usabilidade se tornou um grande ponto de atenção. Segundo, a ferramenta deve ser facilmente acessível em diferentes aparelhos, mesmo que mais antigos. Por isso, concluiu-se que uma ferramenta web, que pudesse ser acessada diretamente pelo navegador, seria a melhor alternativa.

Além disso, o foco do desenvolvimento seria na versão para celular, já que, de acordo com os dados coletados pelo IBGE em 2021, 84.4% da população brasileira possui celular (R7, 2022) para uso pessoal enquanto que os microcomputadores só estão em

Figura 4.1: Página principal do acervo digital, com o catálogo de itens e os filtros disponíveis.



Fonte: A Autora

40.7% dos domicílios brasileiros (NERY; BRITTO, 2022).

4.2 Fluxo geral do projeto

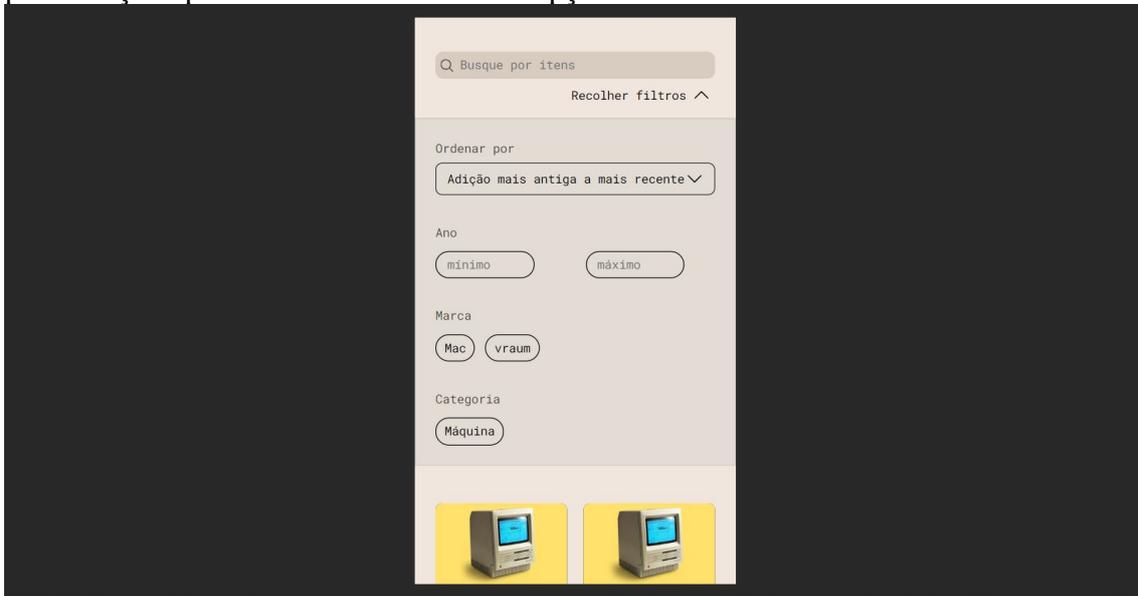
O projeto pode ser dividido em três grandes áreas: o acervo digital, a exploração das exposições físicas e a sessão de administração.

4.2.1 Acervo digital

A sessão de acervo digital, acessível mesmo a usuários não *logados*, contém todos os objetos cadastrados no banco de dados. A figura 4.1 mostra o visual da página. É um catálogo bem simples, onde o usuário pode ver as fotos dos itens com algumas informações e clicar no item para obter mais detalhes. Também há disponível uma barra de busca e alguns filtros, para facilitar a busca por um item mais específico. A figura 4.2 mostra os filtros em mais detalhes.

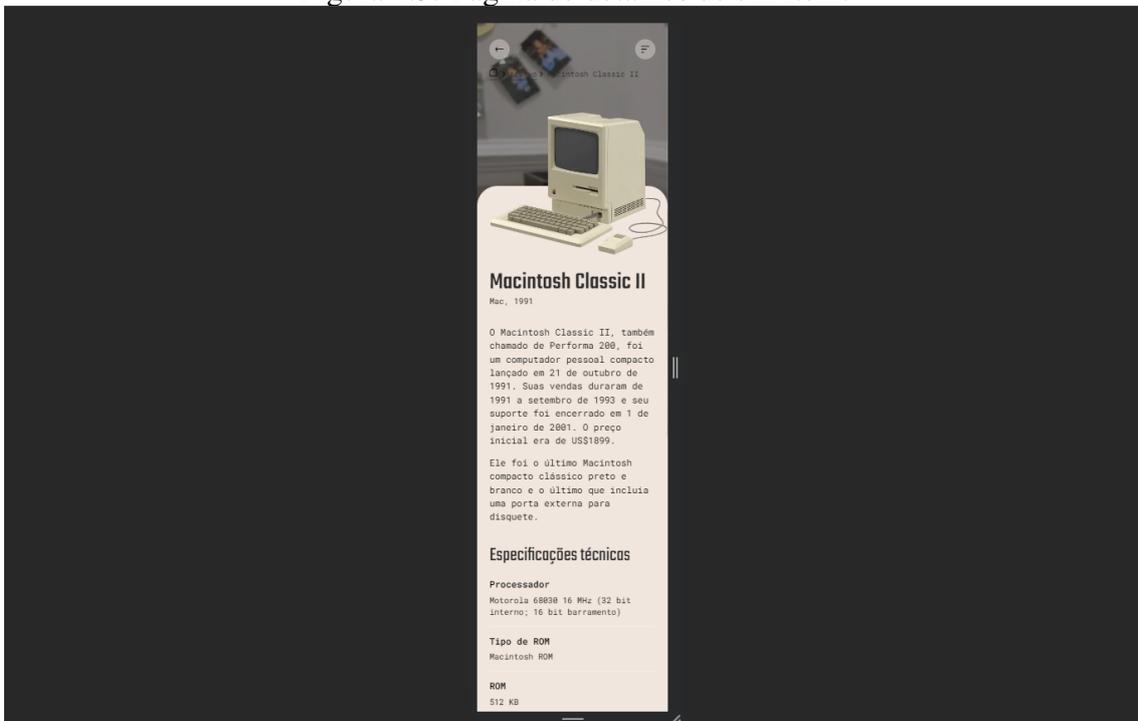
Na página de detalhes de um item, mostrada na figura 4.3, é possível ver uma descrição mais detalhada sobre o item em questão, com descrição, especificações técnicas, fotos e vídeos que abordam o objeto.

Figura 4.2: Os filtros podem ser expandidos. As opções de marca e categoria disponíveis para seleção aparecem de acordo com as opções existentes no banco de dados.



Fonte: A Autora

Figura 4.3: Página de detalhes de um item.



Fonte: A Autora

4.2.2 Exploração das exposições

A exploração das exposições é a principal funcionalidade da aplicação. Nessa seção, é necessário que o usuário possua um registro, que é feito usando a plataforma Google. O registro é necessário para que seja possível armazenar o progresso do usuário ao longo da exploração.

Na exploração das exposições são adotados os conceitos de gamificação abordados no capítulo 3.

4.2.2.1 Escolha da rota

Assim que o usuário entra pela primeira vez, é oferecida a oportunidade de escolher um roteiro de exploração, mas não é obrigatório. O roteiro de exploração tem como objetivo oferecer um direcionamento ao usuário sobre como explorar o Instituto para encontrar as exposições existentes ao criar um roteiro com uma ordem a qual as exposições podem ser encontradas.

4.2.2.2 Desbloqueio de exposição

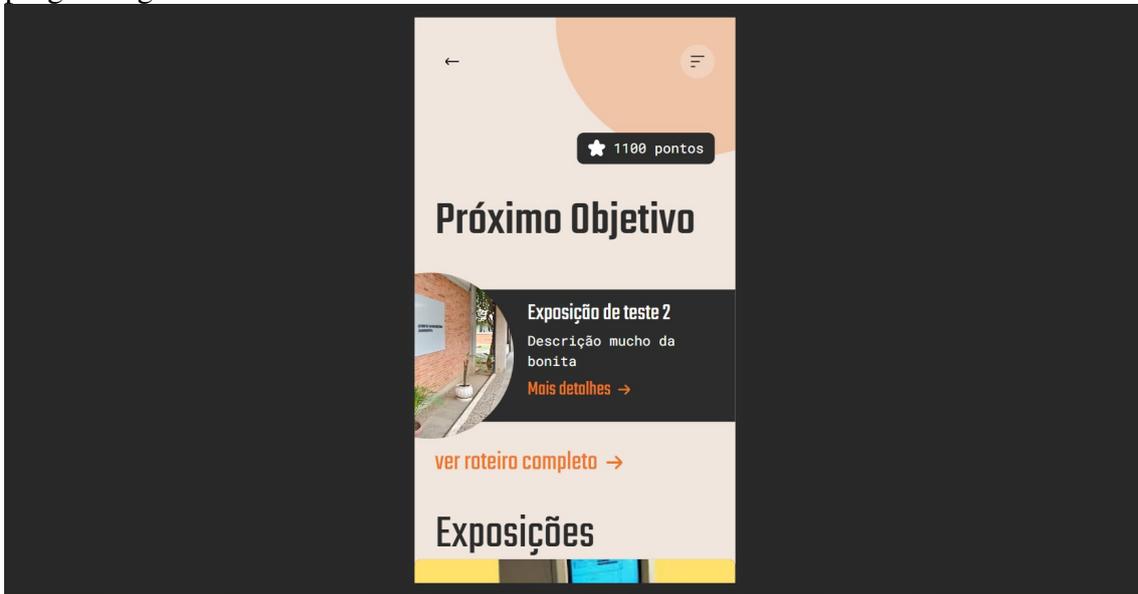
São as missões da aplicação. Toda exposição física tem um *QR Code*, que, ao ser lido, desbloqueia a exposição encontrada para o usuário e o recompensa com pontos e cartas de coleção. O objetivo é estimular o usuário a continuar a busca pelas outras exposições disponíveis.

4.2.2.3 Acompanhamento do próprio progresso

O usuário tem uma página em que pode acompanhar seu progresso geral na exploração. Além da página geral, representada na figura 4.4, o usuário pode acessar páginas com o progresso de cada aspecto individualmente.

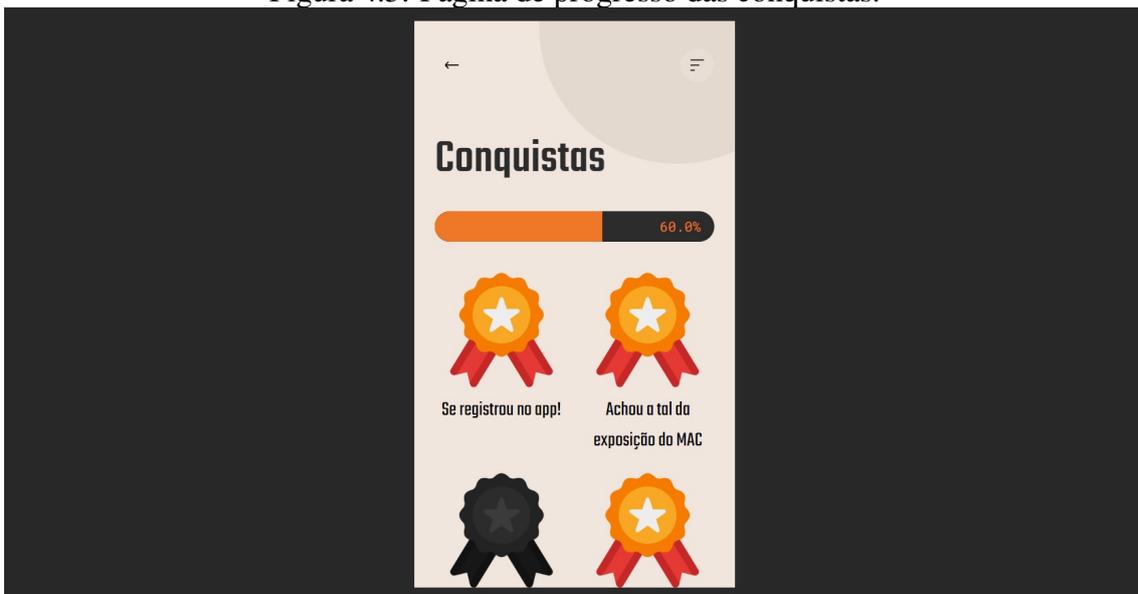
A figura 4.5, por exemplo, mostra a página de progresso das conquistas do usuário. O usuário pode ver seu progresso com relação as conquistas desbloqueadas. Ao clicar na conquista, mostrado na figura 4.6, é possível ver mais detalhes de uma conquista, como sua descrição e a data em que foi desbloqueada.

Figura 4.4: Página geral de progresso do usuário. É possível ver a pontuação atual e seu progresso geral.



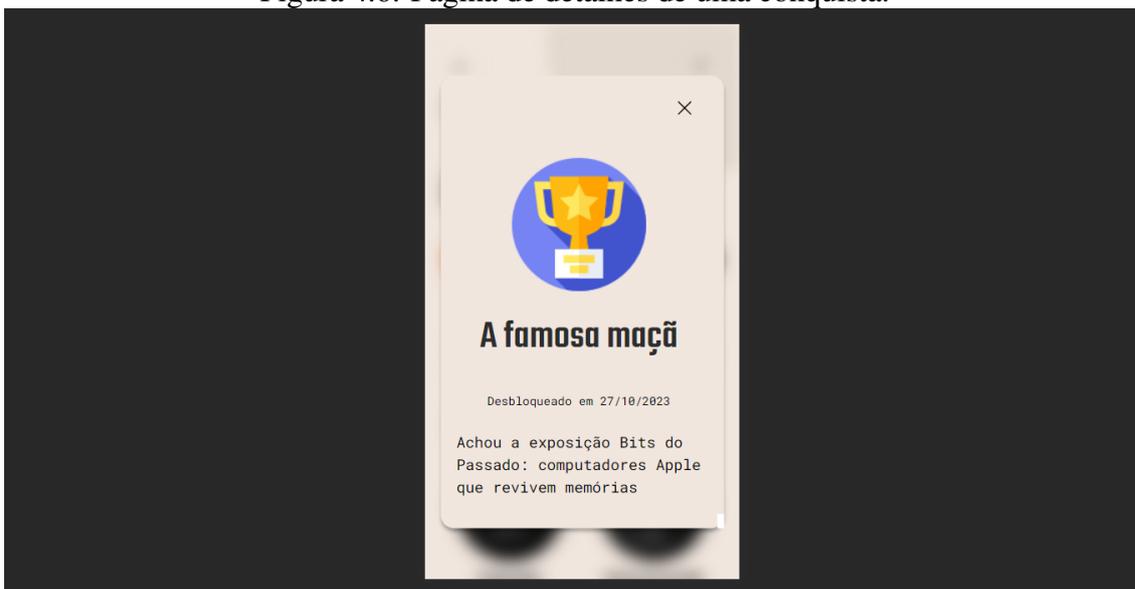
Fonte: A Autora

Figura 4.5: Página de progresso das conquistas.



Fonte: A Autora

Figura 4.6: Página de detalhes de uma conquista.



Fonte: A Autora

4.2.3 Administração

Na seção de administração, acessível apenas por usuários com cargo de administradores, é possível fazer o gerenciamento do conteúdo do banco de dados, sem a necessidade de fazer alterações diretamente no banco e com uma interface mais amigável para usuários não técnicos do que a de plataformas como o Postman, por exemplo.

A interface, representada na figura 4.7 contém um menu lateral, na esquerda, para a escolha do modelo e uma tabela, à direita, com uma listagem de todas as entidades do modelo escolhido. É possível adicionar uma nova entidade e editar ou deletar uma entidade existente.

Na adição, é aberto um formulário com os campos que uma entidade do modelo selecionado pode conter. Os campos obrigatórios, bem como o limite do tamanho de cada campo é marcado visualmente para o usuário. A figura 4.8 mostra o formulário de adição de uma exibição.

Por ser uma seção da aplicação administrativa, com uso de muitas tabelas e formulários, optou-se por fazer essa parte da aplicação apenas na versão *desktop*.

Figura 4.7: Tabela com as informações sobre itens, dentro da página de administração.

The screenshot shows the 'Administração' page with a sidebar menu on the left containing 'Itens', 'Exibições', 'Roteiros', 'Conquistas', 'Cartas', and 'Administradores'. The 'Itens' section is active, displaying a table with columns for 'Nome', 'Marca', and 'Categoria'. Each row includes a trash icon and an edit icon. The footer contains contact information for the Instituto de Informática at the Universidade Federal do Rio Grande do Sul and social media links.

Nome	Marca	Categoria		
Macintosh Classic II	Mac	Máquina		
Macintosh 512K	Mac	Máquina		
Macintosh Performa 475	Mac	Máquina		
iMac G3	Mac	Máquina		
Muda muda	vraum	Máquina		

Fonte: A Autora

Figura 4.8: Formulário para adição de uma exibição.

The screenshot shows the 'Adicionar exibição' form overlaid on the administration page. The form includes fields for 'Nome *', 'Link do banner *', 'Link da imagem *', 'Descrição *', 'Descrição curta', and 'Localização *'. Each text input field has a red border and a placeholder text. The 'Nome' field has a character count of 0/80, 'Link do banner' has 0/100, 'Link da imagem' has 0/100, and 'Localização' has 0/100. The form also features a close button (X) in the top right corner.

Fonte: A Autora

5 MUSEU VIRTUAL: DESENVOLVIMENTO E IMPLEMENTAÇÃO

Neste capítulo será discutido aspectos de desenvolvimento e implementação do museu virtual.

5.1 Arquitetura

A arquitetura do sistema é bem simples e concisa, como mostrado na figura 5.1. O maior foco de desenvolvimento do projeto foi o *frontend*. A parte do *frontend* pode ser dividido em duas áreas principais, as quais ainda podem ser subdivididas em duas áreas. Existe um conjunto de páginas que podem ser acessadas sem a necessidade de realizar o *login* e um conjunto de páginas que podem ser acessadas apenas com *login*.

Dentro das páginas acessíveis sem login, tem a própria página para logar e as outras páginas acessíveis: home, página do acervo e a página de detalhes de cada item. Na sessão de login, é usado a plataforma OAuth do Google para autenticação.

Já na parte acessível por usuários identificados, há as páginas que podem ser acessadas por usuários comuns e uma página que só pode ser acessada por usuários com poderes administrativos.

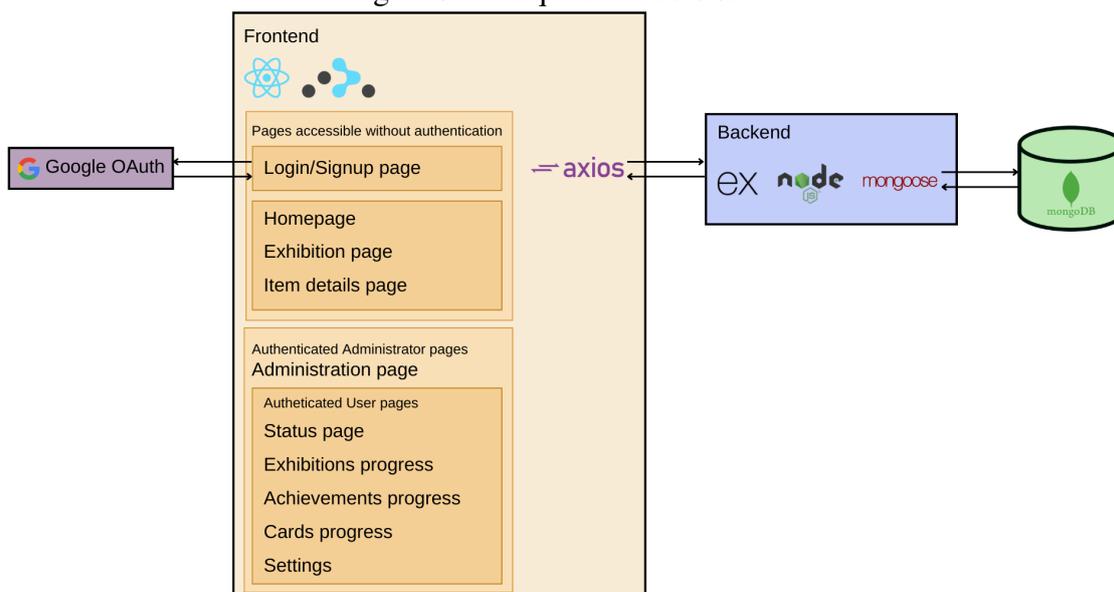
A comunicação entre *frontend* e *backend* é feita através da biblioteca Axios. O *backend*, por sua vez, se comunica com o banco de dados usando a biblioteca Mongoose.

5.1.1 Fluxo de cadastro de usuário

Para exemplificar um fluxo que passe por toda a arquitetura do sistema, será usado o registro de um usuário. Um usuário que deseja se cadastrar no sistema e entrar na página de *login* deve seguir o passo a passo:

1. Na página de *login*, no *frontend*, são mostradas as contas Google cadastradas no navegador do usuário em *pop ups*;
2. Quando o usuário escolhe uma conta, é enviado uma requisição ao Google OAuth pedindo autorização para o acesso do nome, *email* e foto de perfil do usuário;
3. Em caso de sucesso, é retornado ao *frontend* um *token JWT* com as informações do usuário;
4. Nesse momento, o *frontend* utiliza o Axios para fazer uma requisição *HTTP GET*

Figura 5.1: Arquitetura do sistema.



Fonte: A Autora

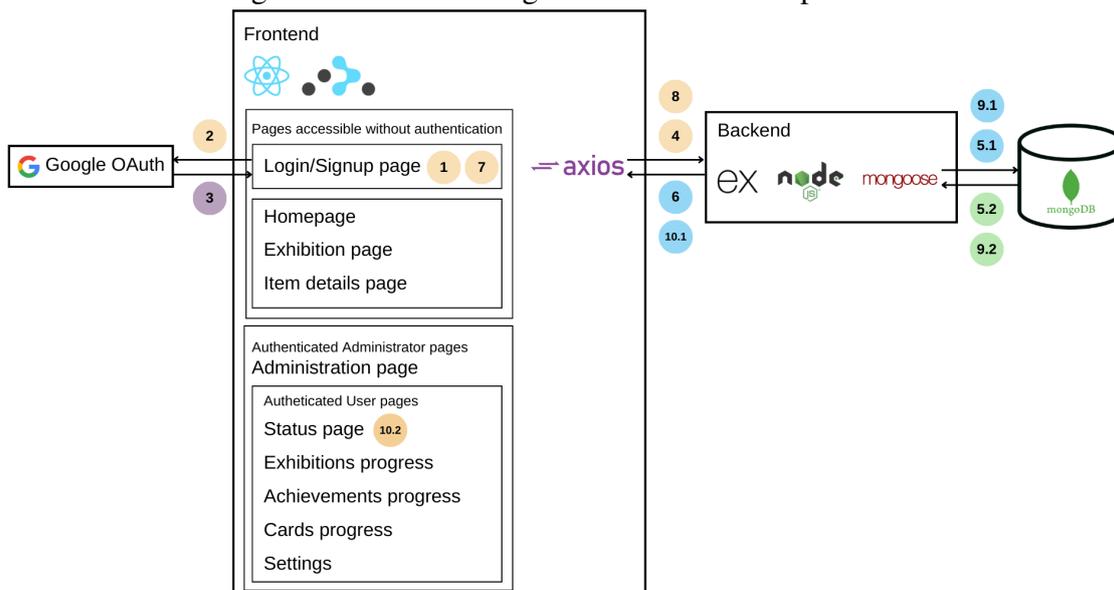
- para o *backend* solicitando a conquista de se registrar na aplicação;
5. No *backend*, utilizando a biblioteca Mongoose, os dados da conquista solicitada são consultados;
 6. O dados são formatados para um formato compatível e enviados ao *frontend*;
 7. Se a requisição for bem sucedida, os dados da conquista e do usuário são estruturados em um objeto, parecido com o formato que será salvo no banco de dados;
 8. O objeto é colocado no corpo de uma requisição *HTTP POST* para o *backend*;
 9. O objeto do usuário é salvo no banco de dados;
 10. Se o registro for bem sucedido, o usuário é redirecionado para a página do Seu Perfil.

O fluxo foi ilustrado na figura 5.2.

5.2 Banco de Dados

Para o banco de dados, foi utilizado *MongoDB* (versão 6.0.6). Foi escolhido um banco não relacional, pois este tem maior flexibilidade com relação aos campos de cada entidade, o que para essa aplicação seria proveitoso.

Figura 5.2: Fluxo do registro do usuário no aplicativo.



Fonte: A Autora

5.2.0.1 Aquisição dos Dados

Os membros do PET foram responsáveis pela separação e classificação dos itens, bem como pela separação dos objetos em grupos (exibições) e coleta das informações sobre cada um. Por ser uma tarefa manual e que demanda um longo tempo, no período de desenvolvimento desse trabalho, não foram catalogados muitos itens.

5.2.1 Construção dos modelos

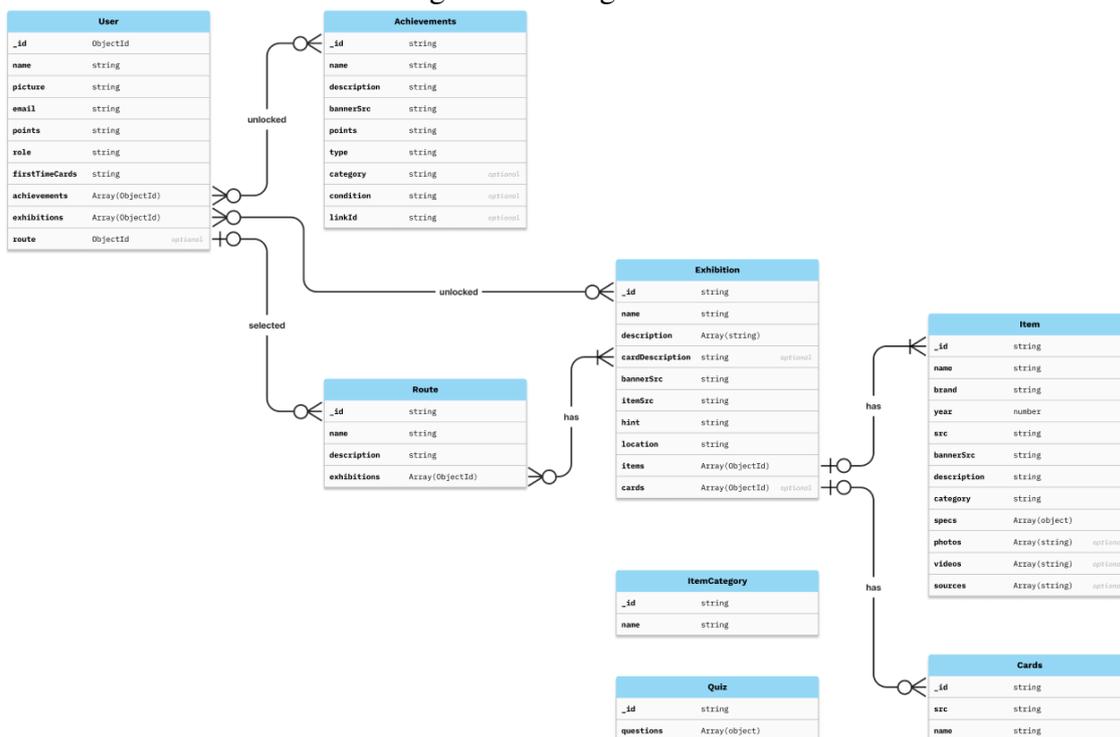
Por ser feito em um banco não relacional, as coleções tem maior flexibilidade com relação aos atributos. Porém, é possível definir atributos fixos para cada coleção, como mostrado na figura 5.3.

Para a modelagem do sistema, foram construídos oito modelos: *achievements*, *cards*, *exhibitions*, *items*, *itemCategories*, *quizes*, *routes* e *user*. Esses modelos serão detalhados a seguir

5.2.1.1 Achievements

O modelo *achievements* representa uma conquista.

Figura 5.3: Diagrama ER.



Fonte: A Autora

5.2.1.2 Cards

O modelo *cards* representa uma carta de uma coleção de cartas. É um modelo simples, contendo apenas um nome, descrição e o uma url para a imagem.

5.2.1.3 Exhibition

O modelo *exhibitions* representa uma exibição presente no Instituto. Além de conter atributos como nome e descrição, ela também guarda quais itens e cartas estão vinculadas a ela.

5.2.1.4 Items

O modelo *items* representa um item catalogado do acervo.

5.2.1.5 Categoria de item

O modelo *itemcategories* representa as possíveis categorias que um item pode estar alocado.

Tabela 5.1: Tabela dos atributos base esperados para o modelo de conquista

<i>Atributo</i>	<i>Tipo</i>	<i>Obrigatório</i>	<i>Descrição</i>
name	String	sim	Nome da conquista. Deve ter no máximo 80 caracteres.
description	String	sim	Descrição da conquista. Deve ter no máximo 200 caracteres.
bannerSrc	String	sim	Link da imagem da conquista. Deve ter no máximo 200 caracteres.
points	String	sim	Quantos pontos a conquista concede. Uma conquista não pode conceder pontos negativos e não pode conceder mais de 50000 pontos.
type	String	sim	Tipo da conquista. Uma conquista pode ser do tipo <i>milestone</i> ou <i>linked</i> .
category	String	sim	Categoria da conquista. A categoria se refere a qual outra entidade/acontecimento a conquista está vinculada.
condition	Number	não	Condição necessária para atingir a conquista.
linkId	String	não	Id da entidade vinculada a conquista.

Fonte: A Autora

5.2.1.6 Quizes

O modelo *quizes* representa um questionário, com suas perguntas e possíveis respostas.

5.2.1.7 Routes

O modelo *routes* representa uma das possíveis rotas que o usuário pode escolher para seguir. É um modelo simples, contendo apenas um nome, descrição e o conjunto de exposições associadas.

5.2.1.8 User

O modelo *users* representa um usuário cadastrado no sistema.

Tabela 5.2: Tabela dos atributos base esperados para o modelo de cartas

<i>Atributo</i>	<i>Tipo</i>	<i>Obrigatório</i>	<i>Descrição</i>
name	String	sim	Nome da carta. Deve ter no máximo 80 caracteres.
src	String	sim	URL com a imagem da carta. Deve ter no máximo 200 caracteres.

Fonte: A Autora

5.3 API

Para a construção da API, foi utilizada a linguagem *Javascript*, o framework *Express* (versão 4.18.2) e a biblioteca *Moongose* (versão 7.5.2).

A API consiste em um CRUD, com uma organização de projeto muito simples. Sua organização está concentrada no arquivo *server.js* e nas pastas *models* e *routes*, ilustrado na figura 5.4. No arquivo *server.js*, é feita a conexão com o banco de dados e são criados todos os *endpoints* da API. A pasta *models* contém o *Schema* de cada modelo citado na subseção acima do banco de dados. Já na pasta *routes*, cada entidade possui as requisições que podem ser feitas.

Além disso, o projeto conta com um arquivo *swagger.json*, com a documentação completa dos *endpoints* da aplicação e um arquivo *requests.postman_collection.json*, com uma coleção do Postman dos *endpoints* para testes. A documentação pode ser acessada nesse *link*.

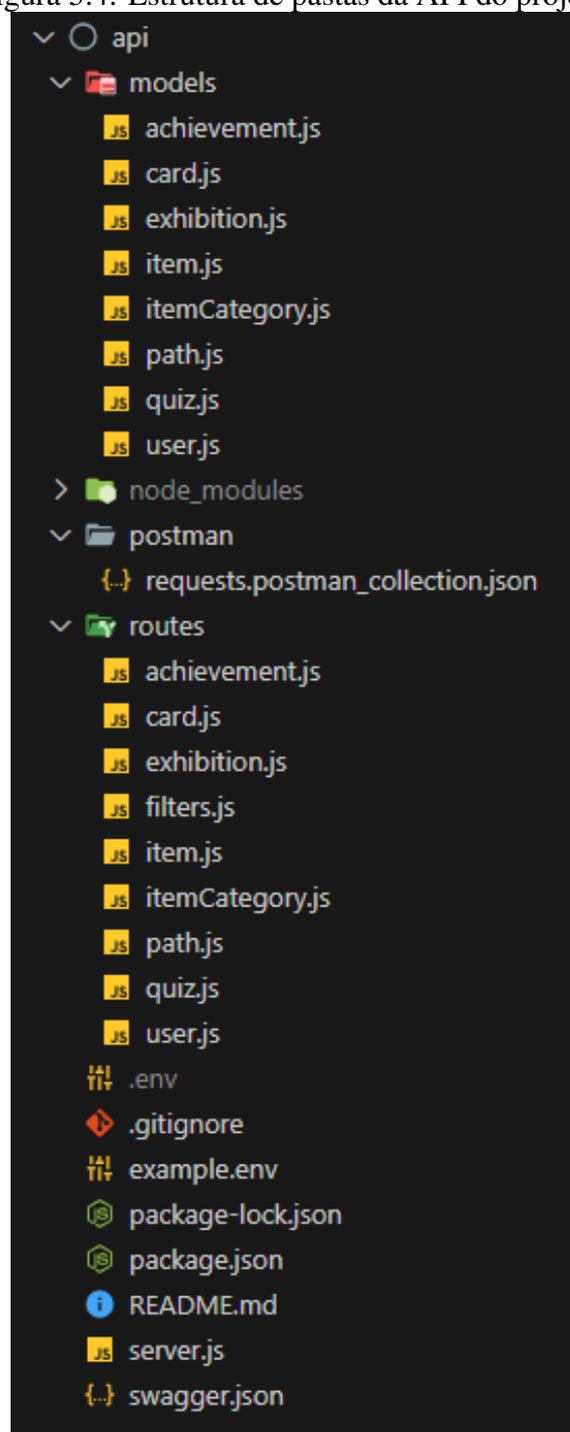
5.4 Frontend

No *frontend*, foi utilizado, principalmente a linguagem *Typescript* (versão 5.2.2) em conjunto com as bibliotecas *ReactJS* (versão 18.2.0), *Axios* (1.5.0), *Router* (6.16.0).

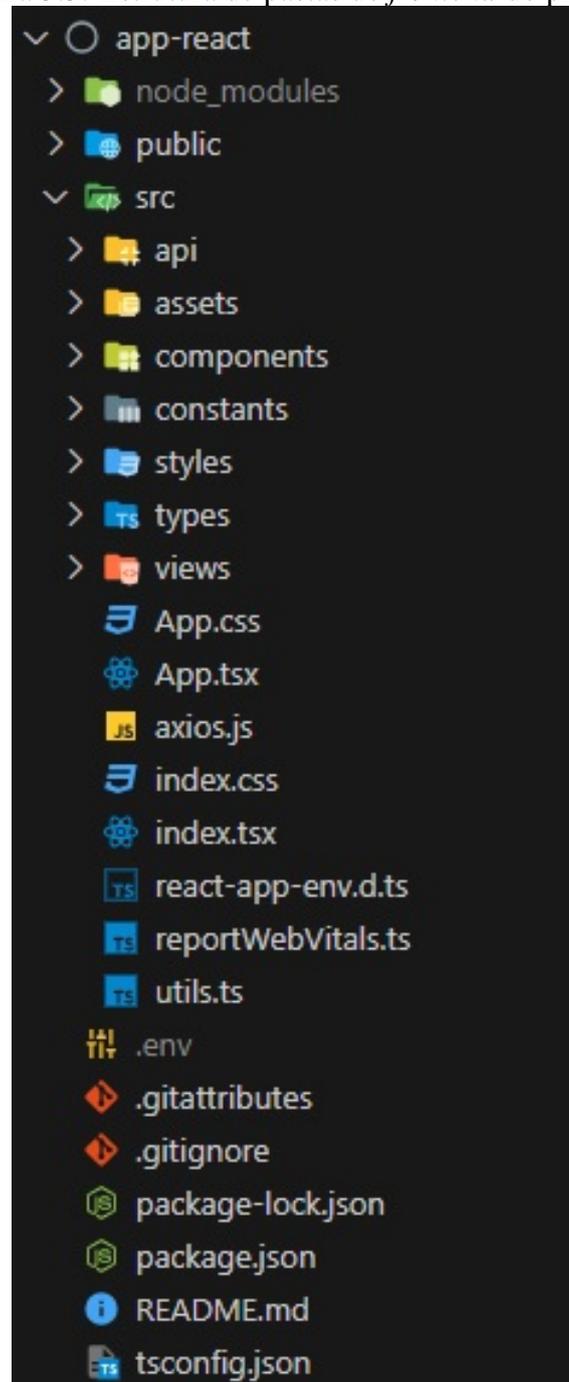
Na parte do *frontend*, a organização, mostrada na figura 5.5 também é bem típica de uma aplicação do gênero. Fora da pasta *src*, onde está contido todo o código principal da aplicação, destacam-se os arquivos *App.tsx*: responsável por organizar todas as rotas da aplicação, além de conter a componente *Footer* e o layout geral que aparece por toda a aplicação; *axios.js*: responsável por fazer a configuração básica de conexão a API e; *utils.ts*: responsável por armazenar algumas funções de uso geral.

A pasta *src*, por sua vez, é organizada nas subpastas *components*, *constants*, *styles*, *types* e *views*. Nas pastas *constants*, *styles* e *types* estão valores constantes, estilos e tipos,

Figura 5.4: Estrutura de pastas da API do projeto.



Fonte: A Autora

Figura 5.5: Estrutura de pastas do *frontend* do projeto.

Fonte: A Autora

Tabela 5.3: Tabela dos atributos base esperados para o modelo de exibição

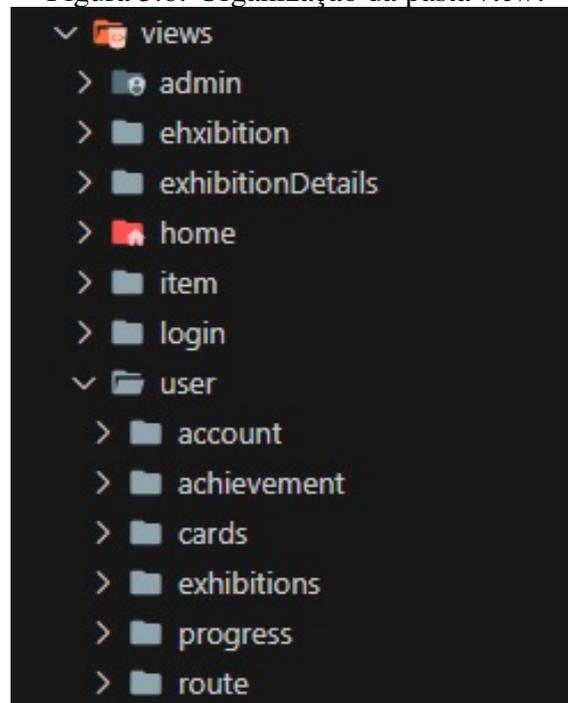
<i>Atributo</i>	<i>Tipo</i>	<i>Obrigatório</i>	<i>Descrição</i>
name	String	sim	Nome da exibição. Deve ter no máximo 80 caracteres.
description	Array	sim	Um vetor de parágrafos contendo uma descrição longa da exibição, que aparece apenas na página de detalhes. Não tem limitação de tamanho.
cardDescription	String	não	Descrição curta da exibição, que aparece nos cards da página principal. Se não for passado a descrição curta, o primeiro parágrafo da descrição longa é usado. Deve ter no máximo 200 caracteres.
itemSrc	String	sim	Link da imagem da exibição. Deve ter no máximo 200 caracteres.
bannerSrc	String	sim	Link do banner da imagem da exibição. Deve ter no máximo 200 caracteres.
hint	String	sim	Um texto com uma dica dada para facilitar encontrar uma exposição ainda não descoberta. Deve ter no máximo 200 caracteres.
location	String	não	O local em que a exibição está localizada fisicamente
items	Array	sim	Lista com os ids das cartas associadas a exibição
cards	Array	não	Lista com os ids das cartas associadas a exibição

Fonte: A Autora

respectivamente, mais genéricos usados ao longo da aplicação.

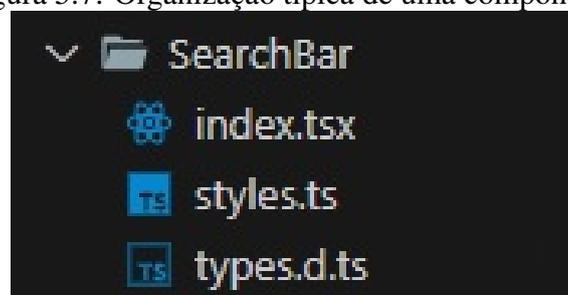
A pasta *views*, contém as páginas presentes na aplicação. Cada pasta corresponde a uma página. Uma página pode ser composta por seções menores, que ficam armazenadas dentro da pasta da página. Essas seções, por sua vez, são compostas por componentes menores. Esses componentes ficam na pasta *components*. A figura 5.6 mostra a organização da pasta.

Uma componente, tipicamente, possui três arquivos, como ilustrado na figura 5.7. O *index.tsx* é responsável por conter toda a lógica da componente, o *styles.ts* faz a estilização e o *types.d.ts* contém as propriedades que a componente precisa receber.

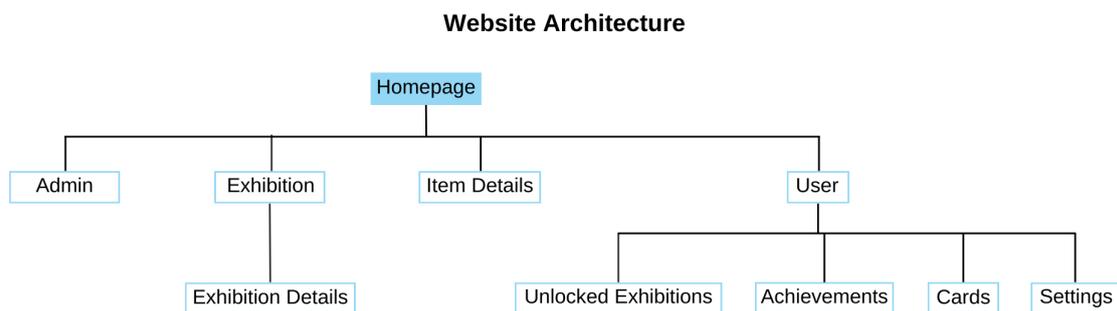
Figura 5.6: Organização da pasta *view*.

Fonte: A Autora

Figura 5.7: Organização típica de uma componente.



Fonte: A Autora

Figura 5.8: Estrutura de páginas do *frontend*.

Fonte: A Autora

Como pode ser observado na figura 5.8, a aplicação não possui muitas páginas e estas estão estruturadas de forma mais horizontal na hierarquia. O objetivo dessa organização foi oferecer uma navegação simples e lógica ao usuário, sem a necessidade de precisar percorrer uma grande árvore.

5.5 Gamificação

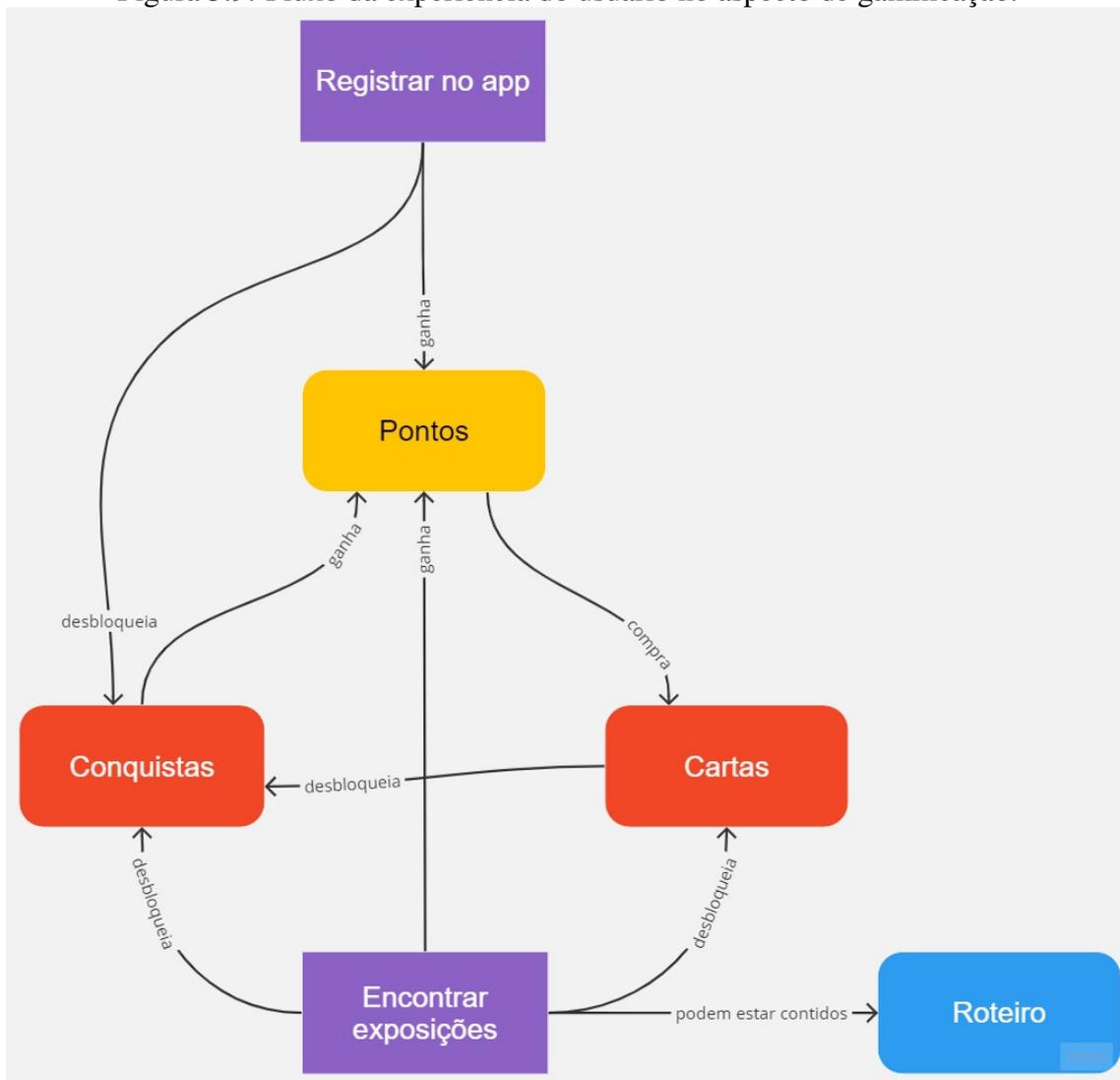
Outro aspecto essencial do desenvolvimento foi tornar a visita mais interativa. A primeira etapa consiste em dividir os itens em grupos, que constituirão as exposições. Essas exposições serão espalhadas pelos prédios do Instituto. Essa distribuição proporciona ao aluno contato com esses objetos históricos durante seu dia a dia na universidade, além de incentivar a exploração por parte dos visitantes dos diferentes locais da universidade. O segundo passo é introduzir alguns conceitos de gamificação.

Nesse contexto, o objetivo é motivar os visitantes a explorarem todas as exposições e enriquecer sua visita física, com informações adicionais que complementam as informações presenciais. Gamificação pode ser feita de várias formas. Para esse trabalho foram pensadas em algumas categorias, destacadas no fluxo 5.9, que serão desenvolvidas nas seções a seguir.

5.5.1 Pontos

Sistemas de pontos são uma funcionalidade bem clássica em jogos e sistemas gamificados e muito familiar para uma grande parte dos usuários. De acordo com Zicherman and Cunningham (2011), um sistema de pontuação é fundamental para qualquer aplicação gamificada, mesmo que não seja visível ao usuário. Nessa aplicação, os pontos serão

Figura 5.9: Fluxo da experiência do usuário no aspecto de gamificação.



Fonte: A Autora

visíveis ao usuário e servirão como moeda de troca para a compra de cartas.

5.5.2 Conquistas

Conquistas, ou *badges*, também são muito comuns em gamificação. Conquistas são uma boa forma de representar de forma visual o progresso do usuário, estabelecendo marcos ou *checkpoints*.

5.5.3 Encontrar exposições físicas

As exposições, como já descrito, são conjunto de peças de uma mesma temática localizadas em algum prédio do Instituto. Ao encontrar uma exposição física, é possível escanear um QR Code e ter acesso a mais informações sobre a exposição e itens. Desbloquear uma exposição concede ao usuário pontos e conquistas. Também é possível ganhar conquistas a cada número de exposições desbloqueadas.

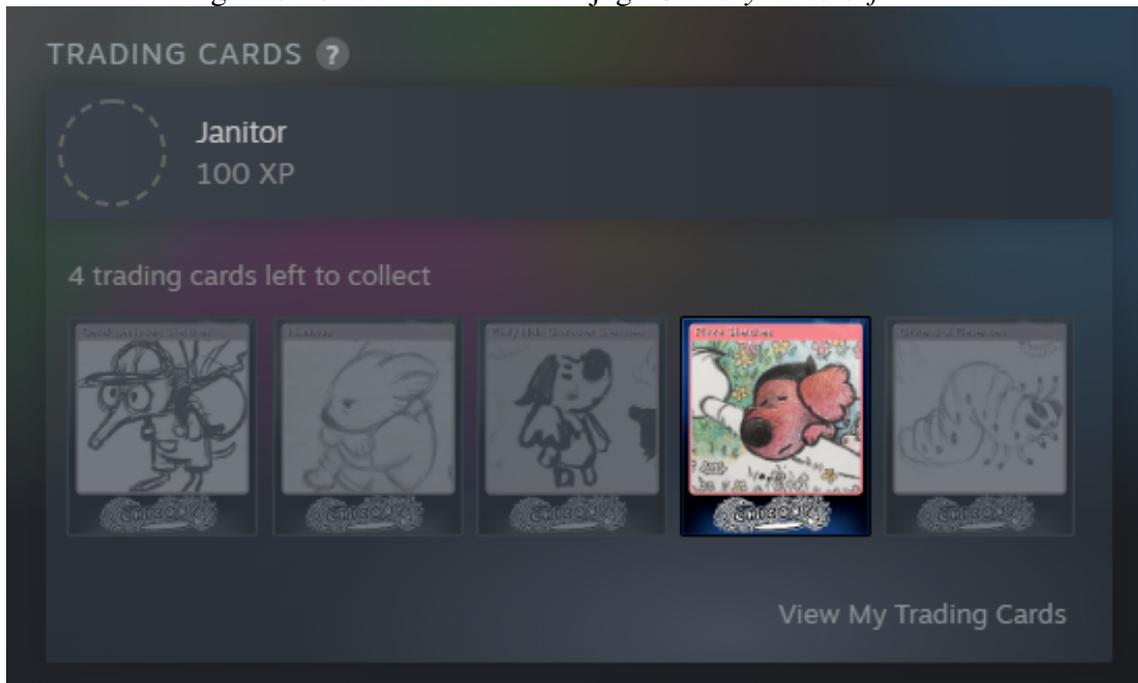
5.5.4 Coleções de cartas

Uma exposição tem ligada a ela um conjunto de cartas. A visita a uma exposição garante algumas das cartas da coleção, mas não todas. Para obter todas as cartas, é necessário comprar as cartas com pontos.

O sistema de coleção de cartas é parecido com o sistema que pode ser encontrado na plataforma de jogos Steam, mostrado na figura 5.10. Nessa plataforma, cada jogo possui um conjunto de cartas e a um determinado número de horas de jogo desbloqueia algumas das cartas da coleção, porém não todas. Para adquirir todas as cartas, é necessário trocar com amigos ou comprar no mercado interno da plataforma.

5.5.5 Roteiro de visitas

Os roteiros têm como objetivo filtrar e ordenar as exposições da melhor forma para cada perfil de usuário. Isso porque nem todos os usuários têm interesse em todos os temas de exposição disponíveis. No entanto, para usuários que preferirem explorar os

Figura 5.10: Cartas de troca do jogo *Chicory: a colorful tale*.

Fonte: A Autora

prédios em busca dos pontos, é possível não escolher roteiro.

5.6 Hospedagem

Nesse primeiro momento, a hospedagem realizada tem como objetivo permitir o teste de usuários, não tendo a intenção de ser a hospedagem permanente para a aplicação.

5.6.1 Banco de Dados

Foi usado o próprio serviço do Atlas para hospedar o banco de dados, não sendo necessária nenhuma configuração.

5.6.2 Backend

Para fazer a hospedagem do backend foi seguido o tutorial de Wizard (2023). Foi criada uma instância EC2 com Ubuntu, o serviço de máquina virtual oferecido pela AWS. Depois de configurado, foi criado um *runner* no Github Actions para realizar o *deploy* da aplicação toda vez que ela fosse alterada em produção. Esse *runner* será responsável pelo

fluxo do CI/CD. As variáveis de ambiente foram armazenadas diretamente no Github.

5.6.3 Frontend

A hospedagem do frontend foi feita seguindo o tutorial de Keys (2023). Foi criado um *pipeline* utilizando o Code Pipeline, serviço oferecido pela AWS para CI/CD. Depois, foi criado um Code Build, outro serviço oferecido pela AWS, para realizar o *build* do código. A hospedagem é feita em um *bucket* no S3, serviço de armazenamento. As variáveis de ambiente foram armazenadas no Parameter Store do AWS.

Tabela 5.4: Tabela dos atributos base esperados para o modelo de item

<i>Atributo</i>	<i>Tipo</i>	<i>Obrigatório</i>	<i>Descrição</i>
name	String	sim	Nome do item. Deve ter no máximo 80 caracteres.
brand	String	sim	Nome da marca do item. Deve ter no máximo 50 caracteres.
year	Number	sim	Ano que o item foi lançado.
description	Array	sim	Um vetor de parágrafos contendo uma descrição do item. Não tem limitação de tamanho.
src	String	sim	Link da imagem do item. Deve ter no máximo 200 caracteres.
bannerSrc	String	sim	Link da imagem do banner do item. Deve ter no máximo 200 caracteres.
category	String	sim	Categoria do item. Deve ser uma das categorias do modelo <i>itemcategories</i>
specs	Array	sim	Vetor de objetos contendo especificações técnicas do item. O objeto deve possuir dois atributos: <i>label</i> e <i>value</i> , ambos do tipo <i>String</i> . <i>Label</i> deve ter no máximo 50 caracteres e, <i>value</i> , no máximo 80.
photos	Array	não	Lista com todas as urls de fotos do item.
videos	Array	não	Lista com todas as urls de vídeos do item.
sources	Array	não	Lista com todas as urls de fontes de referência do item.

Fonte: A Autora

Tabela 5.5: Tabela dos atributos base esperados para o modelo de categoria de item

<i>Atributo</i>	<i>Tipo</i>	<i>Obrigatório</i>	<i>Descrição</i>
name	String	sim	Nome da categoria

Fonte: A Autora

Tabela 5.6: Tabela dos atributos base esperados para o modelo de quiz

<i>Atributo</i>	<i>Tipo</i>	<i>Obrigatório</i>	<i>Descrição</i>
question	Array	sim	Lista das perguntas e possíveis respostas

Fonte: A Autora

Tabela 5.7: Tabela dos atributos base esperados para o modelo de rotas

<i>Atributo</i>	<i>Tipo</i>	<i>Obrigatório</i>	<i>Descrição</i>
name	String	sim	Nome da rota. Deve ter no máximo 80 caracteres.
description	String	sim	Descrição da rota. Deve ter no máximo 200 caracteres.
exhibitions	Array	sim	Lista com os ids das exposições associadas a rota.

Fonte: A Autora

Tabela 5.8: Tabela dos atributos base esperados para o modelo de usuário

<i>Atributo</i>	<i>Tipo</i>	<i>Obrigatório</i>	<i>Descrição</i>
picture	String	sim	Url da foto do usuário vinda do Google. Deve ter no máximo 200 caracteres.
name	String	sim	Nome do usuário vindo do Google. Deve ter no máximo 80 caracteres.
email	String	sim	Email do usuário vindo do Google. Deve ter no máximo 100 caracteres.
role	String	sim	Cargo do usuário no sistema. Ele pode ser um usuário comum ou administrador.
points	Number	sim	Pontos do usuário na aplicação.
exhibitions	Array	não	Lista de objetos que correspondem as exposições já descobertas pelo usuário. Esse objeto contém o id da exposição, o dia em que foi desbloqueada e as cartas adquiridas.
achievements	Array	não	Lista de objetos que correspondem as conquistas já atingidas pelo usuário. Esse objeto contém o id da conquista e o dia em que foi desbloqueada.
firstTimeCards	boolean	sim	Indicativo se o usuário já entrou a sessão de cartas pela primeira vez ou não. Na primeira vez que o usuário entra nessa sessão, é exibida uma mensagem explicando o que são as cartas.
route	String	não	Id da rota escolhida pelo usuário, se existir.

Fonte: A Autora

6 AVALIAÇÃO

Uma avaliação do projeto foi feita com 16 alunos da graduação do Instituto de Informática entre os dias 27 e 28 de novembro de 2023. Como mostrado na figura 6.1, foi selecionado um público-alvo bem específico para realização do teste: alunos na graduação na área de Computação. Essa escolha foi feita pensando que os alunos do Instituto serão as pessoas que terão mais contato com as exposições no dia a dia e, portanto, mais chance de usar a aplicação por um período estendido de tempo.

Foram estabelecidas cinco tarefas a serem cumpridas, com o objetivo de explorar as atividades que um usuário faria no dia a dia utilizando a aplicação. Todas as entrevistas foram feitas de forma presencial, o que permitiu observar as dificuldades dos usuários durante a realização das tarefas e coletar *feedbacks* além dos oferecidos no formulário.

- **Tarefa 1:** Entre na página do acervo digital e utilize os filtros para encontrar o item Macintosh 512K;
- **Tarefa 2:** Faça login na aplicação e escolha um roteiro (de sua preferência);
- **Tarefa 3:** Entre na página da exposição Nome da exposição e utilize a dica para desbloquear a exposição;
- **Tarefa 4:** Compre uma carta aleatória;
- **Tarefa 5:** Altere o roteiro escolhido inicialmente.

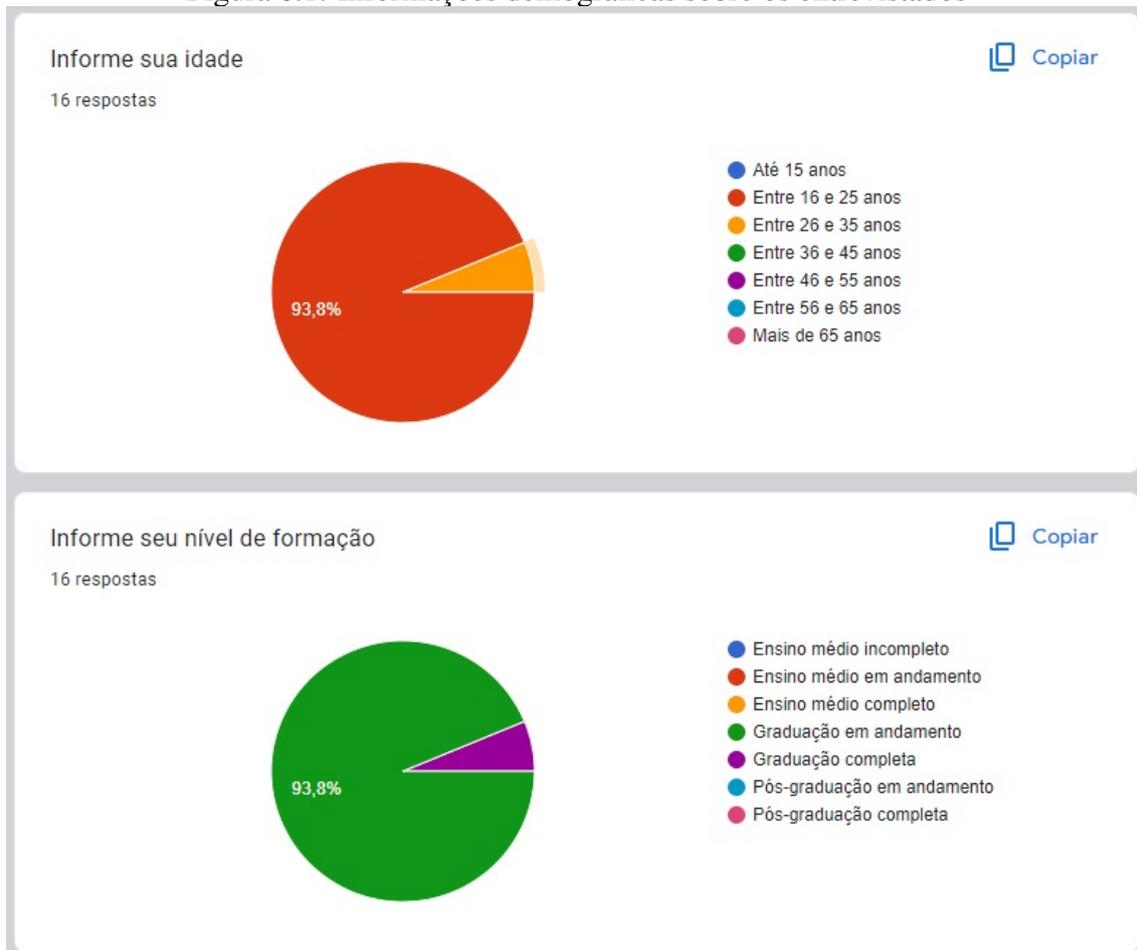
Ao iniciar o processo de avaliação, era explicado ao usuário qual a proposta do trabalho, as tarefas a serem realizadas e o *link* do site. A aluna não interferia no processo de realização das tarefas, mas estava disponível para dúvidas e auxílios eventuais.

As tarefas de um a quatro, de forma geral, não ofereceram grande desafio aos participantes da pesquisa. Na tarefa 1, representada pela figura 6.2, 87,6% dos entrevistados apresentaram pouca ou nenhuma dificuldade. A tarefa 2, gráfico da figura 6.3, teve um percentual bem próximo de facilidade de execução que a tarefa 1, 87,5%.

As tarefas 3 e 4, representadas pelos gráficos 6.4 e 6.5, respectivamente, já apresentaram um pouco mais de dificuldades para os entrevistados, embora ainda tenham tido poucos usuários com grande dificuldade.

Para a tarefa 3, alguns usuários apontaram que acharam difícil encontrar a exposição indicada. Na tarefa 4, por sua vez, não houve nenhum comentário dos usuários que estavam testando, mas uma possível causa para a maior dificuldade pode ter sido encontrar a página onde seria possível comprar cartas, já que não existe uma página "Loja". A

Figura 6.1: Informações demográficas sobre os entrevistados



Fonte: A Autora

Figura 6.2: Avaliação da dificuldade dos usuário ao realizar a tarefa 1, sendo 1 muita dificuldade e 5 muita facilidade.



Fonte: A Autora

Figura 6.3: Avaliação da dificuldade dos usuário ao realizar a tarefa 2, sendo 1 muita dificuldade e 5 muita facilidade.



Fonte: A Autora

Figura 6.4: Avaliação da dificuldade dos usuário ao realizar a tarefa 3, sendo 1 muita dificuldade e 5 muita facilidade.



Fonte: A Autora

compra é feita na página "Suas Cartas".

A tarefa 5, no entanto, apresentou mais dificuldades, como representado na figura 6.5. A principal dificuldade foi encontrar o botão que altera o roteiro dentro da aplicação.

Por fim, a última seção pede um *feedback* geral sobre a usabilidade e interesse na aplicação. As figuras 6.7 e 7.1 demonstram que a aplicação teve uma boa recepção por parte dos usuários, com a grande maioria dos usuários classificando muito positivamente a interface e relatando a experiência como divertida.

É importante destacar que o experimento foi feito com um grupo muito homogêneo, estudantes de computação na graduação. Dessa forma, entende-se que já havia familiaridade por parte dos participantes com websites. A amostragem ser tão homogênea é uma limitação e é alvo para trabalhos futuros.

Figura 6.5: Avaliação da dificuldade dos usuário ao realizar a tarefa 4, sendo 1 muita dificuldade e 5 muita facilidade.



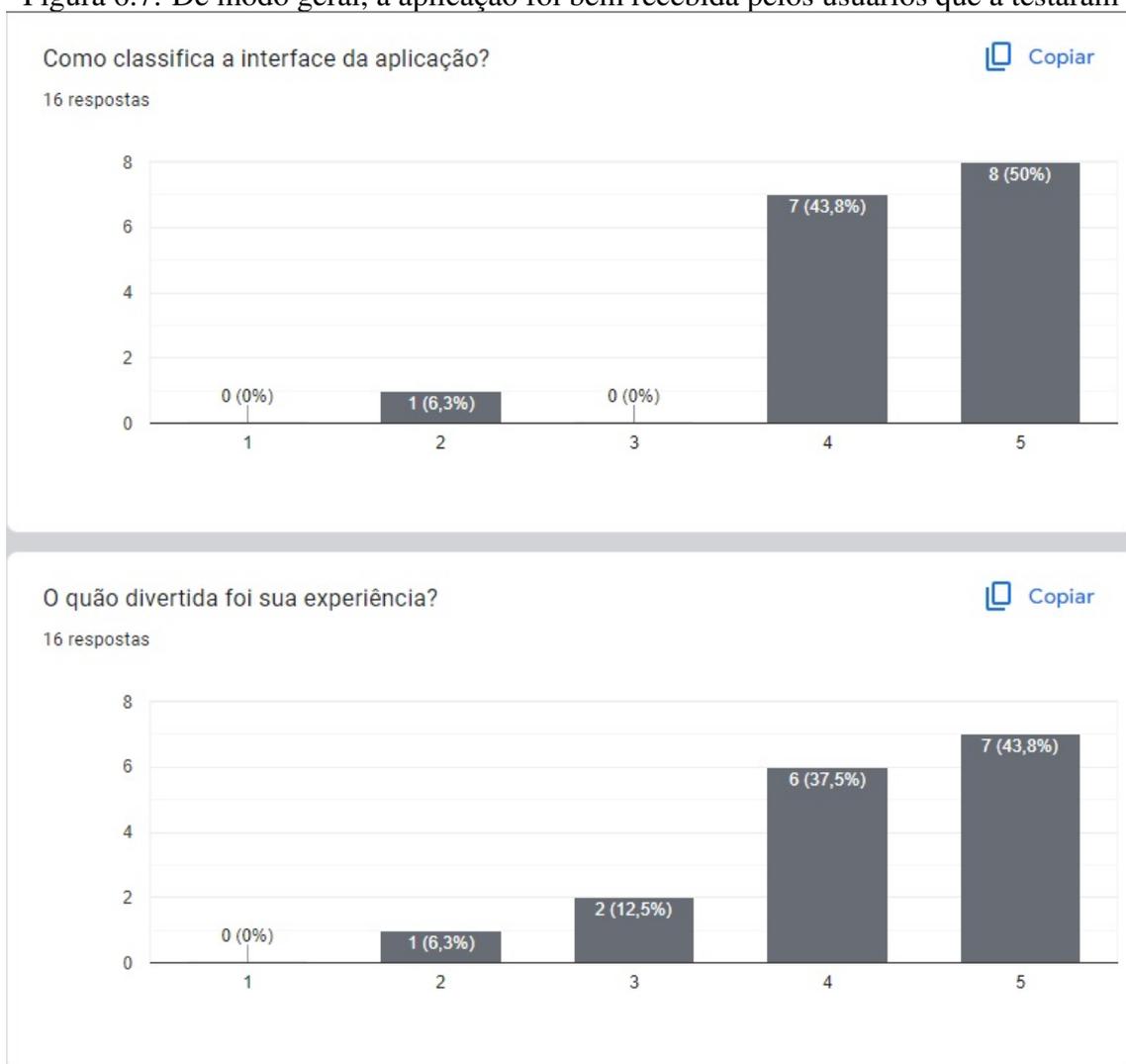
Fonte: A Autora

Figura 6.6: Avaliação da dificuldade dos usuário ao realizar a tarefa 5, sendo 1 muita dificuldade e 5 muita facilidade.



Fonte: A Autora

Figura 6.7: De modo geral, a aplicação foi bem recebida pelos usuários que a testaram



Fonte: A Autora

7 CONCLUSÃO

Nessa seção será discutidos os resultados obtidos, limitações do projeto feito e possibilidades para trabalhos futuros.

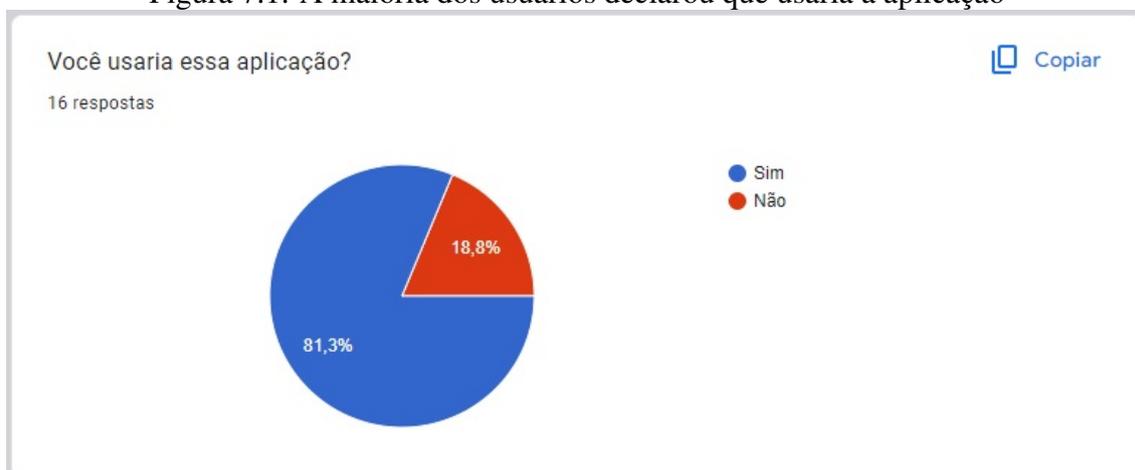
7.1 Resultados obtidos

O presente trabalho tinha como objetivo desenvolver uma plataforma para o INF UFRGS com um acervo digital abrangente, que cataloga os itens físicos presentes no Instituto; e que fizesse o uso de estratégias de gamificação para enriquecer a experiência dos visitantes ao acervo (digital ou físico).

Durante o texto foi apresentado como outros museus adicionaram interatividade às suas exposições e quais aspectos foram usados como inspiração. A partir disso, dado que o tempo para desenvolvimento é curto, foi pensado em um MVP que pudesse testar a efetividade que um aplicativo poderia ter para complementar a experiência física do acervo.

Os resultados obtidos foram muito satisfatórios, dado que 81,3% das pessoas entrevistadas usariam o aplicativo, como mostrado na figura 7.1. Mas, apesar de um retorno positivo dos usuários, ainda são necessários refinamentos para se tornar um produto.

Figura 7.1: A maioria dos usuários declarou que usaria a aplicação



Fonte: A Autora

7.2 Limitações

7.2.1 Login

No projeto apresentado, a única forma de realizar o login é utilizando uma conta Google. O tempo curto para o desenvolvimento do trabalho, não foi priorizado a criação de um sistema de login próprio, já que essa escolha implic em estabelecer medidas segurança, como criptografia.

7.2.2 Funcionalidades não implementadas

Pensando na construção de um MVP, foi necessário deixar diversas funcionalidades interessantes como possibilidades para o futuro. Há diversas adições que poderiam enriquecer a experiência do usuário, entre elas:

7.2.2.1 *Eventos*

Eventos são uma forma de manter a exploração mais dinâmica, incluindo exposições não permanentes. Um exemplo de exposição não permanente que já acontece no Instituto é o museu montado pelo prof. Cabral anualmente no Portas Abertas. A participação nos eventos garantiria pontos e conquistas exclusivas.

7.2.2.2 *Interação entre usuários*

Outra funcionalidade que é muito interessante na gamificação é a interação entre usuários. É possível usar a própria comunidade para incentivar a exploração das exposições. Isso pode ser feito com *leaderboards*, que podem ser semanais, mensais e gerais; oferecer a possibilidade de troca de cartas entre usuários; ou permitir que um usuário dê dicas sobre exposições ainda não desbloqueadas.

7.2.3 Experiência de usuário e navegação

Por fim, outro aspecto que ainda poderia ser melhorado é a experiência de usuário e navegação, garantindo uma navegação mais fluida pela aplicação.

7.2.4 Amostragem

Como dito em seções anteriores, o público alvo da aplicação consiste em alunos do INF, que frequentam regularmente os prédios do Instituto, e o público mais geral, que comparece em eventos organizados pela UFRGS.

Durante o trabalho, a avaliação do MVP desenvolvido foi feita exclusivamente com alunos da graduação, que já têm mais conhecimento sobre tecnologia. Seria de extrema importância avaliar com usuários leigos, que teriam dificuldades e experiências diferentes na aplicação.

7.3 Trabalhos futuros

O trabalho pode ser expandido de diversas maneiras, em diferentes direções. Uma possível melhoria, o caminho de expansão mais direto, seria implementar as funcionalidades citadas na seção anterior. Isso traria uma experiência gamificada ainda mais completa e dinâmica.

Outra trabalho futuro mais direto seria avaliar como pessoas leigas usariam a aplicação desenvolvida, encontrado suas principais dificuldades e avaliando como melhorar a experiência para esse público em particular.

Pensando em uma direção mais voltada para realidade aumentada ou realidade virtual, também seria possível incluir modelagem 3D dos objetos do acervo e visualizá-los dentro da página de detalhes de cada item. Além de poder visualizar o objeto nas três dimensões, também seria enriquecedor criar um museu virtual que tivesse todos esses objetos modelados expostos, permitindo o uso de realidade virtual.

Por fim, outro ponto de expansão poderia ser a generalização da plataforma e a criação de uma plataforma *low code* ou *no code* que permitisse outros museus no Brasil fazerem o uso da plataforma personalizando para seus contextos.

REFERÊNCIAS

AMERICAN MUSEUM OF NATURAL HISTORY. **Explorer App**. Acesso em: 08 de novembro de 2023. Available from Internet: <<https://www.amnh.org/plan-your-visit/explorer>>.

AWS. **O que é a API RESTful?** Acesso em: 08 de novembro de 2023. Available from Internet: <<https://aws.amazon.com/pt/what-is/restful-api/>>.

AWS. **What is a Web Application?** Acesso em: 08 de novembro de 2023. Available from Internet: <<https://aws.amazon.com/what-is/web-application/>>.

AXIOS. Acesso em: 08 de novembro de 2023. Available from Internet: <<https://axios-http.com/ptbr/docs/intro>>.

BARTLE, R. A. **Designing Virtual Worlds**. [S.l.: s.n.], 2003.

BESSA, A. **AWS: Guia sobre o que é Amazon Web Services, seus Serviços e Certificações**. Acesso em: 21 de novembro de 2023. Available from Internet: <<https://www.alura.com.br/artigos/aws>>.

BESSA, A. **Node.JS: o que é, como funciona esse ambiente de execução JavaScript e um Guia para iniciar**. Acesso em: 09 de novembro de 2023. Available from Internet: <<https://www.alura.com.br/artigos/node-js>>.

BURKE, B. **Gamificar: como a gamificação motiva as pessoas a fazerem coisas extraordinárias**. [S.l.]: DVS Editora, 2015.

ESPPENCHUTZ, G. **Introdução ao Mongoose para MongoDB**. 2022. Acesso em: 09 de novembro de 2023. Available from Internet: <<https://www.freecodecamp.org/portuguese/news/introducao-ao-mongoose-para-mongodb/>>.

KEYS, W. O. **AWS Code Pipeline Deploying React app to s3 Bucket (CI/CD) (2020)**. 2023. Acesso em: 18 de novembro de 2023. Available from Internet: <<https://youtu.be/Mgs7jl430vs?si=rYTMwMRyoGCWv3km>>.

KLOCKA, A. C. T. et al. Tailored gamification: A review of literature. **International Journal of Human-Computer Studies**, 2020.

LONGEN, A. **React Descomplicado: O Que é e Como Começar Já!** 2023. Acesso em: 08 de novembro de 2023. Available from Internet: <<https://www.hostinger.com.br/tutoriais/o-que-e-react-javascript>>.

MATA, G. Z. da; MGTV. **Explorer App**. 2018. Acesso em: 08 de novembro de 2023. Available from Internet: <<https://g1.globo.com/mg/zona-da-mata/noticia/2018/09/19/museu-em-juiz-de-fora-usa-qr-code-para-ampliar-experiencia-de-visitantes.ghtml>>.

MONGODB. **MongoDB Atlas Tutorial**. Acesso em: 21 de novembro de 2023. Available from Internet: <<https://www.mongodb.com/basics/mongodb-atlas-tutorial#:~:text=MongoDB%20Atlas%20is%20a%20fully,scale%20MongoDB%20in%20the%20cloud>>.

MONGODB. **O que é o MongoDB?** Acesso em: 09 de novembro de 2023. Available from Internet: <<https://www.mongodb.com/pt-br/what-is-mongodb>>.

MOZILLA. **CRUD**. 2023. Acesso em: 13 de novembro de 2023. Available from Internet: <<https://developer.mozilla.org/pt-BR/docs/Glossary/CRUD>>.

MUSEUS, C. **Conhecendo Museus | Episódio: MUSEU DO AMANHÃ**. 2020. Acesso em: 08 de novembro de 2023. Available from Internet: <<https://youtu.be/hqmQPtteg9U?si=230bQMtx88c56vyU>>.

NERY, C.; BRITTO, V. **Internet já é acessível em 90,0domicílios do país em 2021**. 2022. Acesso em: 10 de novembro de 2023. Available from Internet: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/34954-internet-ja-e-acessivel-em-90-0-dos-domicilios-do-pais-em-2021#:~:text=De%202019%20a%202021%2C%20a,%25%20para%209%2C9%25.>>>

OKTA. **O que é o OAuth 2.0?** Acesso em: 08 de novembro de 2023. Available from Internet: <<https://auth0.com/pt/intro-to-iam/what-is-oauth-2>>.

ORACLE. **O que é NoSQL?** Acesso em: 09 de novembro de 2023. Available from Internet: <<https://www.oracle.com/br/database/nosql/what-is-nosql/>>.

PET COMPUTAÇÃO UFRGS. Acesso em: 08 de novembro de 2023. Available from Internet: <<https://www.inf.ufrgs.br/site/pet>>.

R7. **Mais de 155 milhões de brasileiros possuem celular para uso pessoal, aponta IBGE**. 2022. Acesso em: 10 de novembro de 2023. Available from Internet: <<https://www.correiopovo.com.br/jornalcomtecnologia/mais-de-155-milh%C3%B5es-de-brasileiros-possuem-celular-para-uso-pessoal-aponta-ibge-1.891007>>.

REDHAT. **CI/CD**. 2023. Acesso em: 21 de novembro de 2023. Available from Internet: <<https://www.redhat.com/pt-br/topics/devops/what-is-ci-cd>>.

ROVEDA, U. **TYPESCRIPT: O QUE É, COMO COMEÇAR E QUAIS SÃO AS VANTAGENS?** 2023. Acesso em: 08 de novembro de 2023. Available from Internet: <<https://kenzie.com.br/blog/typescript/>>.

SANTOS, D. **MVP: O que significa e como usar no desenvolvimento de software**. 2018. Acesso em: 09 de dezembro de 2023. Available from Internet: <<https://blog.db1.com.br/mvp-no-desenvolvimento-de-software/?gclid=CjwKCAiAvdCrBhBREiwAX6-6Ukig7sr-zHEz1kvu3rmOX1AmaqH18qyJMSBe4kbyjPV769u4Tg6shBoBwE>>.

SCIENCE MUSEUM UK. **Power Up**. Acesso em: 08 de novembro de 2023. Available from Internet: <<https://www.sciencemuseum.org.uk/see-and-do/power>>.

SOUSA, F. **ReactJS: O básico do React Router**. 2022. Acesso em: 08 de novembro de 2023. Available from Internet: <<https://medium.com/@filipefilpe/o-b%C3%A1sico-do-react-router-b375a5c4bd9b>>.

TV, C. F. **Styled Components (O construtor de elementos do React) // Dicionário do Programador**. 2022. Acesso em: 08 de novembro de 2023. Available from Internet: <<https://youtu.be/p4jYFJNEyqA?si=wxJ171JnDYuYAbJj>>.

WIZARD, W. **Auto Deploy Nodejs Rest API on AWS EC2 | CI/CD pipeline using GitHub Actions**. 2023. Acesso em: 15 de novembro de 2023. Available from Internet: <https://youtu.be/cgWXQqL-ZU8?si=MksG8rH2q28_Pov5>.

ZICHERMAN, G.; CUNNINGHAM, C. **Gamification by Design**. [S.l.]: O'Reilly Media Inc, 2011.