

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ALEXANDRE TAKESHI OGASSAHARA GONZAGA

**E-MUSEU: EXPLORANDO A HISTÓRIA DA INFORMÁTICA EM UM AMBIENTE
VIRTUAL**

GUARAPUAVA

2024

ALEXANDRE TAKESHI OGASSAHARA GONZAGA

**E-MUSEU: EXPLORANDO A HISTÓRIA DA INFORMÁTICA EM UM AMBIENTE
VIRTUAL**

E-Museum: Exploring the history of Computing in a Virtual Space

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Tecnologia em Sistemas para Internet do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná.

Orientadora: Prof^ª. Sediane Carmem Lunardi
Hernandes

Coorientadora: Prof^ª. Angelita Maria De Ré

GUARAPUAVA

2024



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

ALEXANDRE TAKESHI OGASSAHARA GONZAGA

**E-MUSEU: EXPLORANDO A HISTÓRIA DA INFORMÁTICA EM UM AMBIENTE
VIRTUAL**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Tecnólogo em Tecnologia em Sistemas
para Internet do Curso Superior de Tecnologia
em Sistemas para Internet da Universidade
Tecnológica Federal do Paraná.

Data de aprovação: 04/julho/2024

Andres Jessé Porfirio
Doutor em Informática
Universidade Tecnológica Federal do Paraná

Diego Marczal
Doutor em Informática
Universidade Tecnológica Federal do Paraná

Sediane Carmem Lunardi Hernandes
Doutora em Informática
Universidade Tecnológica Federal do Paraná

GUARAPUAVA
2024

RESUMO

O cenário de museus virtuais brasileiros ainda é bastante tímido, especialmente na área de informática. A Universidade Tecnológica Federal do Paraná - Campus Guarapuava (UTFPR-GP) e a Universidade Estadual do Centro-Oeste (Unicentro) possuem projetos de extensão ligados ao lixo eletrônico com muitas peças de informática que podem ser catalogadas para a criação de um museu virtual. Assim, a proposta desse trabalho é enriquecer o cenário nacional de museus virtuais com um website de um museu virtual de informática, o E-museu. Desta forma, informações técnicas, detalhes e curiosidades tornam-se acessíveis com facilidade para todas as pessoas que quiserem conhecer um pouco mais sobre a história de computadores, peças e itens de informática. O E-museu foi desenvolvido visando apresentar uma interface simples, amigável e intuitiva com explicações e curiosidades relevantes para instigar e alimentar a curiosidade daqueles que o visitarem. Com o desenvolvimento deste museu virtual de informática, a história, a cultura e a tecnologia são preservadas.

Palavras-chave: museu virtual; museu virtual de informática; história de computadores; tecnologia da informação; programação para internet.

ABSTRACT

The scenario of brazilian virtual museums is still very timid, specially in the field of computing. The Federal University of Technology - Paraná (UTFPR) and the State University of the Midwest (Unicentro) have extension projects related to electronic waste with many computer items that can be cataloged for the creation of a virtual museum. Thus, the purpose for this work is enrich the nacional scenario of virtual museums with a website of one virtual museum, making more practical and accessible the information to all of the people who want to know a little more about the history of computers, hardware and items of computing. The E-museum, being developed featuring a simple, friendly and intuitive interface with explanations and curiosities relevant to instigate and feed the curiosity of those who visit it. With the development of this virtual computer museum, history, culture and technology are preserved.

Keywords: virtual museum; virtual museum of computing; history of computers; information technology; web development.

LISTA DE FIGURAS

Figura 1 – Captura de Tela do Museu virtual da Coopermiti.	14
Figura 2 – Captura de Tela do Museu virtual da Universidade Estadual de Maringá.	15
Figura 3 – Captura de Tela do museu virtual do Museu Nacional da Computação do Reino Unido.	16
Figura 4 – Captura de Tela da Página de apresentação do Site Wikipedia.	17
Figura 5 – Fluxo de desenvolvimento adotado para o desenvolvimento do E-museu.	23
Figura 6 – Modelagem do banco de dados do sistema E-museu com base no trabalho de Galvão e Bernardes (2011).	29
Figura 7 – Tela inicial do sistema.	30
Figura 8 – Tela onde apresentam-se os itens do site que podem ser organizados por categoria.	30
Figura 9 – Tela onde apresentam-se os detalhes, informações, o visual e a linhas do tempo de determinado item.	31
Figura 10 – Tela que mostra a linha do tempo de processadores.	31
Figura 11 – Tela onde se disponibiliza o formulário de cadastro de novos itens ao sistema.	32
Figura 12 – Tela sobre as entidades envolvidas na criação do sistema.	32

LISTA DE TABELAS

Tabela 1 – Requisitos Funcionais.	27
Tabela 2 – Requisitos Não Funcionais.	27
Tabela 3 – Resultados do teste de Usabilidade.	44

LISTAGEM DE CÓDIGOS FONTE

Listagem 1 – Variáveis de ambiente para a configuração do banco de dados.	34
Listagem 2 – <i>Migration</i> da tabela items do banco de dados.	36
Listagem 3 – <i>Model</i> de <i>Proprietary</i>	37
Listagem 4 – <i>Factory</i> de itens.	38
Listagem 5 – <i>Seeder</i> de <i>Section</i>	39
Listagem 6 – Fragmento de uma <i>View</i> em <i>Laravel Blade</i>	41
Listagem 7 – Fragmento de diálogo da assistente.	41
Listagem 8 – Função <i>store</i> de <i>AdminCategoryController</i>	42
Listagem 9 – Função <i>store</i> de <i>AdminCategoryController</i>	43

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Objetivos	10
1.1.1	Objetivo geral	10
1.1.2	Objetivos específicos	10
1.2	Justificativa	11
1.3	Estrutura do trabalho	11
2	REVISÃO DE LITERATURA	12
2.1	Museu virtual	12
2.2	Trabalhos Relacionados a Museus virtuais	13
2.2.1	Museu virtual da Coopermiti	14
2.2.2	Museu virtual da Universidade Estadual de Maringá	15
2.2.3	Museu virtual do Museu Nacional da Computação do Reino Unido	16
2.2.4	Wikipedia	17
2.3	Comparação entre os trabalhos relacionados	18
3	MATERIAIS E MÉTODOS	19
3.1	Materiais	19
3.1.1	Laravel	19
3.1.2	Laragon	20
3.1.3	Visual Studio Code	20
3.1.4	Bootstrap	20
3.1.5	jQuery	21
3.1.6	DbSchema	21
3.1.7	MySql	22
3.1.8	Figma	22
3.1.9	Clip Studio Paint	22
3.1.10	Git e Github	22
3.2	Métodos	23
3.2.1	Fluxo de Desenvolvimento	23
3.2.2	Levantamento de Requisitos	24

3.2.3	Modelagem do Banco de Dados	24
4	RESULTADOS ALCANÇADOS	25
4.1	Escopo do sistema	25
4.2	Modelagem do sistema	25
4.2.1	Requisitos Funcionais e Não Funcionais	26
4.2.2	Banco de Dados	28
4.3	Apresentação do sistema	29
4.4	Implementação do sistema	34
4.4.1	Criação do Sistema	34
4.4.2	Configuração do Sistema	34
4.4.3	Criação das Migrations	35
4.4.4	Implementação de Models	36
4.4.5	Implementação de Factories	37
4.4.6	Implementação de <i>Seeders</i>	38
4.4.7	Implementação de Views	39
4.4.8	Implementação de Controllers	41
4.4.9	Implementação de Rotas	42
4.5	Implantação do E-museu	43
4.6	Teste de usabilidade do E-museu	43
5	CONSIDERAÇÕES FINAIS	45
5.1	Trabalhos Futuros	46
	REFERÊNCIAS	48
	APÊNDICE A QUESTIONÁRIO DO SUS (SYSTEM USABILITY SCALE)	52

1 INTRODUÇÃO

Desde muito tempo atrás, o homem, por ser curioso e ter interesse no passado, coleciona objetos que fazem parte da cultura e do passado de um povo. Muitas vezes, esses objetos não possuem valor monetário, mas um valor histórico imensurável (BAUER, 2022). Por conta disso, foram criados os museus, locais sem fins lucrativos que tem como objetivo apresentar, toda ou em partes, a história e a cultura de um povo. Por exemplo, o Museu Britânico¹, um dos maiores museus do mundo, tem como um dos seus artigos a Pedra de Roseta, um item muito importante para a história da civilização egípcia, que já foi uma das maiores do mundo.

Um museu não precisa ser um local físico e pode ser criado no mundo digital. Isso é conveniente, pois a Internet está acessível a uma grande parte da população (SILVA, 2022). Com isso, facilita a visita de pessoas que, talvez, não teriam acesso se o museu fosse físico. O museu do Louvre², o maior museu de arte do mundo, disponibiliza a visita de quatro galerias de arte, através da Internet, com muitas informações adicionais nas obras. A NASA³ também possui algo similar em que o usuário pode se locomover virtualmente pelos seus centros de pesquisa, analisar objetos e ferramentas, além de contar com textos informativos sobre cada detalhe.

Aprender sobre o passado é compreender o presente, tanto em questões culturais como também em questões tecnológicas, de forma a evitar erros cometidos. Dentro deste contexto, os museus de informática existem para servir a esse mesmo propósito, mostrar para as pessoas o passado dos computadores, ensinar e registrar tudo o que foi criado de mais marcante até o momento atual (VOLTOLINI, 2015).

A Universidade Tecnológica Federal do Paraná (UTFPR) e a Universidade Estadual do Centro Oeste (Unicentro) possuem projetos ligados ao lixo eletrônico. A UTFPR possui o projeto Tecno-Lixo: Oficina do Aprender (HAYNE *et al.*, 2022) e a Unicentro o projeto E-Lixo (Ré; THOMEN, 2021). Os projetos recebem peças de computadores que as pessoas não utilizam mais. Muitas peças são reaproveitadas ou reutilizadas nos projetos e várias outras são recolhidas pela empresa Superintendência de Unidades de Conservação, Biodiversidade e Emergências Ambientais (SUC-Ambiental)⁴ por não possuírem mais nenhum uso prático nos projetos. A empresa, recebendo as peças não utilizadas faz o descarte correto das mesmas. A SUC-Ambiental possui licença municipal para isso. Contudo, por um lado, tem-se o descarte correto das peças, mas por outro, a história da peça é perdida; pois, as informações sobre essa peça ajudariam a contar um pouco da história da informática.

Com base no que foi descrito, é importante que algo seja proposto para que a história de algumas peças de informática não sejam perdidas. Nesse sentido, o desenvolvimento

¹ <https://www.britishmuseum.org/>

² <https://www.louvre.fr/en>

³ <https://oh.larc.nasa.gov/oh/>

⁴ Comércio de varejo com foco em peças e acessórios de aparelhos eletrônicos.

de um museu virtual voltado para a área de informática atende ao interesse da sociedade na preservação histórica desses itens tecnológicos.

1.1 Objetivos

Os objetivos deste trabalho se dividem em objetivo geral e objetivos específicos.

1.1.1 Objetivo geral

O objetivo geral do trabalho foi desenvolver um sistema web de um museu virtual de informática com o propósito registrar e apresentar para as pessoas, a história e aparência de computadores, peças de computadores ou itens de informática.

1.1.2 Objetivos específicos

Os objetivos específicos definidos para o trabalho foram:

- implementar um sistema web com uma interface amigável (fácil de usar) e intuitiva (fácil de entender);
- possibilitar o cadastro, atualização e remoção de peças de computador e itens de informática pelo usuário administrador do sistema;
- permitir que usuários do website sugiram e ajudem a cadastrar novas peças no sistema;
- possibilitar que o administrador do sistema web analise e aceite, ou não, as sugestões de novas peças feitas pelos usuários ou adição de curiosidades aos itens já cadastrados;
- organizar os itens do museu no website por categoria e também por data;
- adicionar imagens e detalhes em texto para cada item do museu;
- desenvolver uma página com todos os itens de uma determinada série, ordenados por data;
- criar uma assistente virtual para auxiliar na navegação, contextualizar páginas e oferecer instruções de uso do sistema;
- desenvolver um sistema de navegação com um visual que simule uma linha do tempo para os itens do museu;
- realizar testes de usabilidade para verificar a eficácia da interface.

1.2 Justificativa

A criação desse museu virtual de informática abre espaço para apresentar a história de itens variados de informática que foram recebidos pelos projetos E-Lixo (Ré; THOMEN, 2021) e Tecno-Lixo: Oficina do Aprender (HAYNE *et al.*, 2022). Estes itens, que foram doados, por não terem mais utilidade, podem ganhar um novo propósito.

Desta forma, é de interesse histórico, social e cultural, o desenvolvimento de um museu virtual voltado para a área de informática, que registre e apresente para as pessoas, a história e aparência de computadores, peças de computadores ou itens de informática que um dia já foram utilizados. Como consequência, o impacto que a tecnologia teve em nossa sociedade, facilitando a execução de tarefas rotineiras no cotidiano das pessoas é demonstrado.

1.3 Estrutura do trabalho

O trabalho é dividido como segue. No Capítulo 2 são descritos os conceitos principais sobre Museus virtuais e os trabalhos relacionados. O Capítulo 3 trata dos materiais e métodos que foram utilizados para o desenvolvimento do E-museu, incluindo a motivação para a escolha de cada uma das ferramentas. No Capítulo 4 são apresentados o escopo e os requisitos do sistema, a apresentação das telas do E-museu, detalhes do processo do seu desenvolvimento, implantação e o teste de usabilidade adotado e aplicado para avaliação da usabilidade do website. Por fim, no Capítulo 5 algumas considerações finais e sugestões de trabalhos futuros são descritas.

2 REVISÃO DE LITERATURA

Neste capítulo serão apresentados alguns conceitos de museu virtual e alguns trabalhos relacionados.

2.1 Museu virtual

A definição de museu virtual é um assunto ainda bastante controverso. Diversos amantes e estudiosos de museologia discutem sobre e dão suas próprias conclusões a respeito. Rosali Henriques fez um estudo e pesquisou sobre a opinião de dezenas de pessoas envolvidas no âmbito museológico. A autora não concordou com a maioria das definições dadas pelos pesquisados, pois se apoiavam muito na ideia de que um museu virtual era apenas um ambiente virtual para se expor o objeto material digitalizado. Após ponderar sobre, chegou a seguinte conclusão: "(...) Museu virtual é aquele que faz da internet espaço de interação através de ações museológicas com o seu público"(HENRIQUES, 2018).

Com isso, entende-se que não basta catalogar e listar os itens de um museu com uma descrição extensa de seu funcionamento. A chave para que um museu possa ser considerado como tal é de como é apresentada essa informação. Baseando-se nessas questões, o Museu virtual de informática deste projeto, o E-museu, apresenta computadores, peças de computadores ou itens de informática de forma clara ao usuário, com explicações e curiosidades que realmente são relevantes para instigar e alimentar a curiosidade daqueles que visitarem o museu.

Em Dutra (2018), Eichler e Pino (2007), Schweibenz (1998) quatro tipos de museus virtuais são apresentados, sendo que todos possuem características similares, porém se diferenciam por quais informações e pelo modo que são apresentadas ao público. São eles:

- **Museu Panfleto/Folheto:** Tem como princípio divulgar o museu, apresentando suas exposições e artigos que poderão ser visualizados somente no museu físico no qual o panfleto faz referência.
- **Museu de Conteúdo:** O conteúdo do museu é apresentado de maneira mais direta e convida o visitante a explorá-lo de maneira online. Por não ter didática na apresentação das coleções do museu, esse estilo de museu é mais recomendado aos visitantes que já possuem um conhecimento prévio sobre o que vão ver.
- **Museu Interativo:** Apresenta o museu e suas coleções de maneira mais didática, muitas vezes utilizando uma linguagem que seja eficaz para a idade e para o conhecimento prévio dos visitantes. Outro ponto interessante é a presença de informações adicionais para os itens do museu que ajudam a instigar a curiosidade de seus visitantes.

- **Museu virtual:** São os museus que possuem informações detalhadas de suas coleções. Suas informações são escritas e apresentadas com bastante qualidade, sendo muitas vezes escritas por profissionais da área. Alguns museus virtuais se baseiam em museus físicos, e apresentam exposições que possam ter sido descontinuadas no museu físico.

Analisando os tipos de museus é possível chegar a conclusão de que o Museu virtual de informática que foi desenvolvido nesse projeto tem muitas similaridades com o museu Interativo e o museu virtual. Isso porque o E-museu contempla a presença de descrições em uma linguagem mais simples e pensada para ser lida por pessoas de todas as idades e grupos sociais, informações adicionais para alimentar e instigar a curiosidade dos visitantes, informações detalhadas e aprofundadas àquelas pessoas mais interessadas. Além da presença de profissionais da área da tecnologia administrando as informações que serão inseridas ou retiradas do sistema do E-museu.

2.2 Trabalhos Relacionados a Museus virtuais

Não existem muitos exemplares brasileiros de museu virtual. Os poucos que existem estão desatualizados, principalmente pela falta de manutenção e de acompanhamento dos avanços das tecnologias utilizadas para o desenvolvimento de sistemas para a Internet. Desta forma, para ilustrar este cenário foram selecionados alguns museus virtuais, como o Museu virtual da Coopermiti, o Museu virtual da Universidade Estadual de Maringá (UEM) e o Museu Nacional da Computação do Reino Unido. Também foi incluído um exemplo não classificado como museu virtual, a Wikipedia.

2.2.1 Museu virtual da Coopermiti

A Coopermiti é uma cooperativa brasileira de triagem de lixo eletrônico sem fins lucrativos, e através disso também visa realizar outros trabalhos, como: inclusão social e digital, educação e capacitação. No site da Coopermiti¹ é disponibilizada uma variedade de páginas e uma delas é a página do museu². Nessa página, apresentada na Figura 1, é possível ter acesso a história, características e, em certas ocasiões, imagens de uma variedade de aparelhos eletroeletrônicos de diferentes fabricantes. Isso, por meio de uma interface simples, porém intuitiva e fácil de navegar. A interface é baseada em uma grade de várias imagens, em que cada imagem representa um item do museu e, é possível obter mais informações sobre o item em questão. Entretanto, há um problema na apresentação de informações, em que todas as informações do item são agrupadas em uma mesma área em texto único, o que pode confundir o usuário no momento da leitura.



Figura 1 – Captura de Tela do Museu virtual da Coopermiti.

¹ <https://coopermiti.com.br>

² <https://coopermiti.com.br/museu-menu/>

2.2.2 Museu virtual da Universidade Estadual de Maringá

Outro exemplo de museu virtual é o museu da UEM, o qual é apresentado na Figura 2. No site do Departamento de Informática (DIN)³ da UEM tem-se o acesso ao museu⁴ em que várias informações sobre o projeto e a área de Tecnologia da Informação (TI) são disponibilizadas, com destaque ao museu representado por uma linha do tempo. Sua interface é bem simples e defasada por não ter sido atualizada há bastante tempo, porém é muito rica em informações, não só de aparelhos eletroeletrônicos, mas também da história da computação.

MUSEU DO COMPUTADOR - UEM

Museu HTML

Pré-História
a.C. | 1622 a 1818 | 1820 a 1898
1905 a 1942

Era dos Computadores
1946 a 1959 | 1960 a 1969 | 1970 a 1979 | 1980 a 1989
1990 a 2004

1970 a 1979

[IBM System/3] [FACT 6501] [GD 71] [IBM 370] [TI 960] [UNIVAC 90/30]

No período compreendido entre 1970 e 1979 o hardware foi caracterizado fundamentalmente pela adoção do "modelo" IBM 360 pela maioria dos fabricantes e pelo lançamento de computadores multiposto com periféricos conectados em rede - nomeadamente Ethernet -.

A própria IBM apresenta publicamente, em 1971, o seu "modelo IBM 360" denominado IBM 370.

IBM 370
1971

O IBM 370 integra a tecnologia de circuito monolítico na construção da memória, em substituição da tecnologia de ferrilha utilizada no IBM 360. Esta tecnologia permite aumentar a capacidade de memória para 262 K. No entanto, esta capacidade de memória é já insuficiente para o eficaz processamento de aplicações e a IBM adopta a tecnologia de memória virtual através do Sistema Operacional DOS/VSE - não confundir com a sigla DOS da Microsoft -.

CPU IBM 370/145

O IBM 370 utilizava os mesmos periféricos que já eram usados no IBM 360, nomeadamente os discos IBM #2314, o que permite a "migração" de um para outro Sistema.

Unidades de disco IBM #2314

Figura 2 – Captura de Tela do Museu virtual da Universidade Estadual de Maringá.

³ <http://ws2.din.uem.br/museu/>

⁴ <http://ws2.din.uem.br/museu/virtualhtml/index.htm>

2.2.3 Museu virtual do Museu Nacional da Computação do Reino Unido

O terceiro exemplo a ser apresentado é o Museu Nacional da Computação do Reino Unido⁵, que oferece o *tour* virtual 3D. A Figura 3 apresenta uma das páginas do site desse museu. O site apresenta, no idioma inglês, informações sobre o museu, como horário de funcionamento e notícias, além da possibilidade de agendamento para visitas presenciais e compras de ingressos para acesso ao museu físico. O site também oferece um *tour* virtual⁶. Nele é possível se locomover para pontos específicos do museu e olhar para todas as direções; e, assim analisar os itens para obter mais informações. O seu funcionamento é bastante similar ao Google Maps⁷ (ferramenta da Google de visualização de mapas do planeta Terra) no modo *Street View* (recurso do Google Maps em que é possível visualizar a localidade do mapa em vistas panorâmicas que simulam três dimensões, facilitando o entendimento daquele local).



Figura 3 – Captura de Tela do museu virtual do Museu Nacional da Computação do Reino Unido.

⁵ <https://www.tnmoc.org>

⁶ <https://www.tnmoc.org/3d-virtual-tour>

⁷ <https://www.google.com.br/maps/>

2.2.4 Wikipedia

Como último exemplo, tem-se a enciclopédia virtual Wikipedia⁸, que, embora não seja denominada um museu virtual, serve para um propósito similar. Dois aspectos deste sistema foram influencias para o desenvolvimento do projeto do E-museu: o primeiro sendo a interface, como mostrado na Figura 4, e o segundo, o aspecto de comunidade e contribuição que os usuários da interface podem realizar. A interface do sistema, apesar de simples, é bastante intuitiva e similar às antigas enciclopédias. Nela, tem-se informações adicionadas pelos próprios usuários do sistema. Como descrito no próprio site da Wikipedia, "*Wikipedia is a free online encyclopedia that anyone can edit, and millions already have.*"(WIKIPEDIA, 2024). Traduzindo ficaria: Wikipedia é uma enciclopédia online gratuita onde qualquer pessoa pode editar, e milhões já o fizeram. Tanto profissionais quanto pessoas sem formação específica podem adicionar suas contribuições e com esse recurso cria-se uma situação com um lado positivo e outro negativo. Por um lado há uma expansão da gama de informações, mas, por outro, nem sempre essas informações são verídicas. E, como se trata de um sistema de proporções colossais, não há uma administração precisa do conteúdo inserido ou retirado.

The screenshot shows the Wikipedia article for "Telefone celular". At the top, there is a search bar and the Wikipedia logo. The article title is "Telefone celular" with a language selector set to "151 línguas". Below the title, there are tabs for "Artigo" and "Discussão". A note indicates that "Celular" redirects to this article. A warning box states that the page cites sources but does not cover all content, and a link is provided to help insert references. The main text discusses mobile phone technology, mentioning the first generation (1G) in the 1980s, the second generation (2G) in the late 1980s and early 1990s, and the third generation (3G) in the late 1990s. A small chart titled "Evolução dos telefones celulares" shows a downward trend in the number of mobile phones over time. The sidebar on the left contains navigation options like "Início", "História", "Operadoras", and "Utilidade". The sidebar on the right contains appearance settings like "Texto", "Pequeno", "Padrão", "Grande", "Largura", and "Largura".

Figura 4 – Captura de Tela da Página de apresentação do Site Wikipedia.

⁸ <https://www.wikipedia.org>

2.3 Comparação entre os trabalhos relacionados

Cada um dos quatro itens apresentados possuem suas particularidades, pontos positivos e também negativos. Como pontos positivos podem-se destacar a interface simples e intuitiva de se navegar do Museu virtual da Coopermiti; a abundância de informações sobre a história da computação e sobre os aparelhos eletroeletrônicos da UEM e a possibilidade de se ter uma experiência remota bem próxima da que seria presencialmente no Museu Nacional da Computação. Como pontos negativos podem-se destacar a falta de investimento ou de atualizações na maneira de apresentar as informações no Museu virtual da Coopermiti e da UEM, apesar desses museus estarem bem completos no quesito informação histórica. No site do Museu Nacional da Informática do Reino Unido, a interface de navegação pelo museu virtual é bastante poluída de informações. E, além disso, há também a barreira da língua, que por ser um site para falantes da língua inglesa, acaba afastando alguns visitantes. Já na Wikipedia, a falta de precisão nas informações lá encontradas é um ponto negativo.

Com base nos pontos positivos e negativos destacados anteriormente, criou-se um sistema web de museu virtual que oferece uma apresentação e navegação são mais simples e amigável para o visitante. Ainda, uma assistente virtual é disponibilizada para auxílio na navegação pelo website, a qual serve como uma guia na navegação. Com a assistente virtual, perguntas básicas são respondidas, ajudando o visitante. Além disso, o sistema facilita a criação e adição de novos artigos no museu (aparelhos ou itens de informática) para o administrador e também para os usuários do site. Isso para que sempre seja incluído conteúdo novo e atualizado.

3 MATERIAIS E MÉTODOS

Neste capítulo, serão apresentadas e brevemente explicadas as principais características que motivaram a escolha dos materiais e métodos utilizados na execução do projeto.

3.1 Materiais

A seguir, os materiais e ferramentas utilizados no desenvolvimento do sistema serão explicados, dando prioridade nas vantagens que apresentam e menos ênfase em informações históricas.

3.1.1 Laravel

O Laravel¹ é um *framework* de desenvolvimento web de código aberto, escrito em PHP por Taylor B. Otwell. Lançado em 2011, foi criado para ser um concorrente mais moderno do CodeIgniter², o qual é amplamente utilizado e suportado por uma comunidade ativa que mantém e auxilia outros desenvolvedores (MELO, 2021). A escolha do *framework* se deu devido a variedade de recursos gratuitos disponíveis para uso em conjunto com ele, além de facilitar a criação, buscas e operações feitas no banco de dados. Algumas outras de suas características que merecem destaque e breve detalhamento são:

- **Utiliza o padrão *Model-View-Controller* (MVC):** Esse padrão de design separa as responsabilidades em três componentes distintos (GUEDES, 2023):
 - **Modelo:** Responsável por manipular os dados da aplicação.
 - **Visão:** Apresenta as informações para o usuário.
 - **Controlador:** Processa os dados enviados pelos usuários e comunica-se com as outras camadas.
- **Escalabilidade:** Laravel pode se adaptar facilmente a futuras mudanças e demandas relacionadas à carga de trabalho (LARAVELHOLDINGSINC, 2024).
- **Facilidade de uso e leitura:** O *framework* possui uma sintaxe simples e elegante, facilitando o desenvolvimento e a manutenção do código (LARAVELHOLDINGSINC, 2024).

Outro fator que motivou a escolha do *framework* foi a familiaridade que o autor possui com a ferramenta, o que tornou o desenvolvimento do sistema foi rápido e eficiente.

¹ <https://laravel.com>

² <https://www.codeigniter.com>

3.1.2 Laragon

O Laragon³ é um ambiente de desenvolvimento para aplicações web que suporta várias linguagens, incluindo *PHP: Hypertext Preprocessor* (PHP), Node.js, Python, Java, Go e Ruby. Foi criado por Leo Khoa, líder em DevOps, para uma empresa de *Software as a Service* (SaaS), ou seja, uma empresa que oferece serviços através da Internet (LARAGON, 2024). Laragon é uma opção leve e de alta performance, podendo utilizar menos de 4MB de memória RAM quando em execução. Além disso, oferece grande flexibilidade ao suportar diversas linguagens de programação e é extremamente fácil de usar, pois as pré-configurações necessárias para a criação de um ambiente de desenvolvimento web são realizadas automaticamente. A partir de sua interface simples e bem organizada, tem-se acesso a opções como, por exemplo, iniciar as funcionalidades do ambiente, visitar o localhost do website, acessar o banco de dados, abrir o terminal de comandos e gerenciar os arquivos do programa e seus projetos. Essas características tornam o Laragon uma ferramenta que acelera significativamente o processo de desenvolvimento de aplicações (PESSOA, 2023).

3.1.3 Visual Studio Code

O Visual Studio Code⁴ é uma *Integrated Development Environment* (IDE), ou seja, um ambiente de desenvolvimento integrado desenvolvido pela Microsoft e lançado em 2015 (REDHATINC, 2023). Trata-se de uma aplicação que reúne diversas ferramentas e recursos necessários para o desenvolvimento de um sistema em uma única interface, tornando o processo mais prático e organizado (HANASHIRO, 2021). Desta forma, com o Visual Studio Code foi possível utilizar um formatador de código, automatizando o processo de edição de código tornando o código mais legível sem a necessidade de investir muito tempo nessa tarefa manualmente. Além disso, o Visual Studio Code possui ferramentas para a verificação de erros e possíveis melhorias no código.

3.1.4 Bootstrap

O Bootstrap⁵ era inicialmente conhecido como Twitter Blueprint, pois foi desenvolvido para ser utilizado no Twitter pelos desenvolvedores Mark Otto e Jacob Thornton. Atualmente, é a biblioteca mais popular para o desenvolvimento de aplicações web utilizando componentes *Cascading Style Sheets* (CSS) (LIMA, 2023). Seu uso permite a criação de sistemas web responsivos a diversas dimensões de tela e interações, além de proporcionar uma aparência agradável. A biblioteca oferece diversos recursos pré-fabricados que, sem necessidade de con-

³ <https://laragon.org>

⁴ <https://code.visualstudio.com>

⁵ <https://getbootstrap.com>

figurações adicionais, podem proporcionar uma boa experiência ao usuário. Utilizar o Bootstrap não significa abdicar da originalidade ou da possibilidade de personalizar o seu sistema. É possível usar CSS próprio em conjunto com a biblioteca para modificar e aperfeiçoar a experiência do usuário na sua aplicação. A instalação do Bootstrap é bastante simples. Seguindo os procedimentos disponíveis na página oficial da biblioteca, basta adicionar duas linhas de código na tag <head> da página da aplicação, e o Bootstrap estará pronto para uso (BOOTSTRAPTEAM, 2024).

3.1.5 jQuery

O jQuery⁶ foi criado em 2006 por John Resig. Licenciado pelo *Massachusetts Institute of Technology* (MIT) e pela *GNU General Public License* (GPL), trata-se de uma API (*Application Programming Interface*), ou seja, uma interface de programação de aplicações. Basicamente, é uma biblioteca JavaScript com diversos recursos úteis para manipulação de *HyperText Markup Language* (HTML), simplificando processos como animação e requisições *Asynchronous JavaScript And XML* (AJAX) (SOUZA, 2021). O Bootstrap utiliza o jQuery em muitos dos seus componentes, tornando necessário o uso dessa biblioteca JavaScript no sistema. O jQuery facilita a seleção de classes e IDs para realizar animações e manipular dados enviados pelos usuários. Um dos recursos do jQuery mais utilizados no sistema foi o AJAX. Com ele, é possível realizar requisições ao banco de dados de forma assíncrona, permitindo que a página web seja atualizada com novos dados dinamicamente, sem a necessidade de recarregamento. Isso é útil para verificar informações fornecidas pelo usuário e fornecer *feedback* em tempo real (AMAZONWEBSERVICESINC, 2024).

3.1.6 DbSchema

O DbSchema⁷ é um software desenvolvido pela companhia alemã Wise Coders, com foco em design e gerenciamento de bancos de dados de forma fácil e intuitiva (WISECODERSGMBH, 2024b). Ele oferece suporte ao uso de mouse para realizar operações, gerar dados aleatoriamente para testes e até usar engenharia reversa para criar o esquema do banco de dados a partir de códigos SQL, permitindo uma visualização clara da estrutura do banco. Essas funcionalidades tornam o DbSchema uma ferramenta poderosa e acessível para os desenvolvedores da área de bancos de dados (WISECODERSGMBH, 2024a).

⁶ <https://jquery.com>

⁷ <https://dbschema.com>

3.1.7 MySql

O MySQL⁸ é um sistema de gerenciamento de banco de dados desenvolvido pela empresa sueca MySQL AB e atualmente pertencente à Oracle . Trata-se de um banco de dados relacional e de código aberto que utiliza a linguagem *Structured Query Language* (SQL) para manipulação de dados. A ferramenta facilita e agiliza o processo de armazenamento e gerenciamento de dados.

3.1.8 Figma

O Figma⁹, desenvolvido por Evan Wallace e Dylan Field, é uma ferramenta de design e prototipação de interface de usuário (*User Interface* (UI)) e experiência de usuário (*User Experience* (UX)). Por ser gratuita, é uma das plataformas mais populares desse gênero, sendo rica em recursos e com funcionalidades que facilitam a colaboração em tempo real com outras pessoas (FIGMA, 2024).

3.1.9 Clip Studio Paint

Desenvolvido pela empresa japonesa CELSYS¹⁰, o Clip Studio Paint¹¹ é um software de ilustração e animação, cuja primeira versão foi lançada em 2001 (CELSYSINC, 2024). Este software é rico em recursos e possui uma vasta comunidade de usuários que constantemente cria e compartilha ferramentas adicionais. Não se trata de um programa, o Clip Studio Paint oferece opções de planos mensais, além da possibilidade de compra única. Este software foi escolhido devido a familiaridade do autor com a ferramenta, já que eu possuía uma licença e tinha conhecimento prévio sobre seu funcionamento e uso.

3.1.10 Git e Github

O Git é um sistema de controle de versão, ou seja, tem a função de controlar e identificar mudanças feitas em arquivos, auxiliando no acompanhamento e organização do trabalho, especialmente quando há mais de uma pessoa envolvida no desenvolvimento. Com o sistema de ramificações, cada desenvolvedor pode editar o código e trabalhar em diferentes funcionalidades sem que as modificações de um interfiram no trabalho do outro. Quando o trabalho em uma ramificação é concluído, as mudanças podem ser mescladas, garantindo que não sejam criadas inconsistências (L., 2024).

⁸ <https://www.mysql.com>

⁹ <https://www.figma.com/pt-br/about/>

¹⁰ <https://www.celsys.com/en/>

¹¹ <https://www.clipstudio.net/pt/>

O GitHub¹² é um host para repositórios Git, ou seja, é um repositório em nuvem onde os códigos relacionados a projetos podem ser armazenados, compartilhados e desenvolvidos em conjunto com outros desenvolvedores. Cerca de 100 milhões de desenvolvedores utilizam a plataforma, com mais de 420 milhões de projetos criados e hospedados (GITHUBINC, 2024).

3.2 Métodos

Nesta seção são apresentadas as etapas de planejamento para a programação do sistema em si.

3.2.1 Fluxo de Desenvolvimento

O fluxo de desenvolvimento adotado para a implementação do sistema web do E-museu é ilustrado na Figura 5.

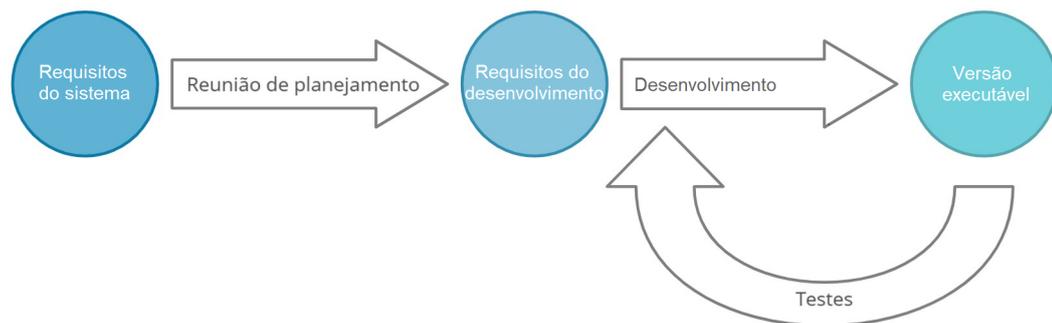


Figura 5 – Fluxo de desenvolvimento adotado para o desenvolvimento do E-museu.

De acordo com a figura, no início foram elaborados os requisitos do sistema, os quais estão relacionados aos objetivos específicos do projeto. Esses requisitos foram levantados baseando-se em outros museus virtuais já consolidados, nos trabalhos de Eichler e Pino (2007), Galvão e Bernardes (2011), além da combinação de entrevistas, *brainstorming*. Após isso, uma reunião de planejamento aconteceu para que os requisitos da fase de desenvolvimento fossem levantados. Os requisitos desta fase são os objetivos a serem alcançados no ciclo de desenvolvimento (no caso deste projeto em específico, o período de desenvolvimento definido inicialmente foi de duas a quatro semanas). Em seguida, a etapa de desenvolvimento aconteceu, onde a programação do sistema foi realizada, a qual teve como base os objetivos estabelecidos na fase anterior. A duração de cada fase de desenvolvimento foi de duas semanas. Ao final de cada fase, uma versão do sistema executável ficou disponível para testes. A maioria dos testes foi realizada pelo próprio desenvolvedor, embora outras pessoas realizaram testes no sistema (Sediane Carmem Lunardi Hernandes - a orientadora do projeto, Angelita Maria De Ré - a coorientadora e alguns alunos da UTFPR e Unicentro). Para coletar o *feedback* (opiniões

¹² <https://github.com>

e sugestões de melhoria a respeito de algum assunto) dos testes, foi aplicado um formulário com algumas perguntas a respeito da usabilidade, baseados nas pesquisas de Forçan *et al.* (2021). Isso para que melhorias na interface fossem desenvolvidas. Ao término de cada ciclo de desenvolvimento, uma reunião com a orientadora e coorientadora era realizada; e, após cada reunião, um novo ciclo de desenvolvimento iniciava com os requisitos para a próxima fase de desenvolvimento atualizados.

3.2.2 Levantamento de Requisitos

Para o desenvolvimento de um software, uma das etapas iniciais e mais importantes é o levantamento de requisitos funcionais e não funcionais. A abordagem escolhida para esse procedimento foi a combinação de entrevistas, *brainstorming* e análise de museus virtuais já consolidados (NATAL, 2018). Alguns fatores contribuíram para essas escolhas, como o número reduzido de pessoas a serem entrevistadas, no caso, duas pessoas, e a simplicidade do projeto, ao considerar outros sistemas similares como referência (mencionados na seção 2.2). Junto com a entrevista, foi possível também utilizar o *brainstorming* para chegar a conclusões mais claras a respeito do escopo do site, bem como definir os requisitos funcionais e não funcionais.

3.2.3 Modelagem do Banco de Dados

Como base para a modelagem do banco de dados foram utilizados os requisitos planejados no levantamento realizado anteriormente na subseção 3.2.2 a esta etapa. Para organizar e filtrar itens de forma que o usuário encontre o que precisa no sistema, considerou-se essencial categorizar os itens do museu virtual de maneira precisa e flexível. Assim, elaborou-se um sistema de etiquetas, onde muitas etiquetas, com nome e subcategoria, poderiam ser relacionadas aos itens, proporcionando uma categorização mais concisa.

4 RESULTADOS ALCANÇADOS

Este capítulo apresenta o sistema do E-museu, o museu virtual de informática, desenvolvido neste trabalho.

4.1 Escopo do sistema

O sistema web do E-museu gerencia processos de um museu virtual de informática, desde o cadastro de peças e acessórios de informática até a apresentação das informações sobre sua história, linhas do tempo, detalhes técnicos e imagens dessas peças. O acesso dos usuários do museu deve ser feito por meio de um website, incluindo a possibilidade de acesso por celular e/ou tablet. Os usuários poderão sugerir itens para o E-museu descrevendo sua especificação técnica, história e curiosidades, associar etiquetas, além de poder fazer upload da foto do item. Desde o início do desenvolvimento deste projeto, optou-se por não exigir que os usuários criassem contas para registrar itens no museu. O objetivo dessa decisão é não desencorajar a colaboração, evitando adicionar uma etapa desnecessária ao processo. Considerando que os dados não são sensíveis e não podem ser modificados por outros usuários após o envio, essa abordagem mostrou-se adequada.

A arquitetura do sistema é baseada em MVC, em que o Laravel é responsável pelo *back-end* (onde está o servidor e a maior parte lógica do site) e por renderizar o *front-end* (o visual e toda a parte em que o usuário pode interagir com o site) para o cliente. O Laravel, também apresenta outras funcionalidades, como, por exemplo, facilitar a conexão do site com o banco de dados, gerenciar rotas, implementar autenticação de usuários, além de oferecer recursos para manter o sistema web mais seguro. As tecnologias utilizadas foram o HTML, CSS e JavaScript no *front-end* do sistema web, e PHP e SQL no *back-end* do sistema web.

O website apresenta um visual simples e fácil de se navegar, com uma assistente virtual que pode ajudar em algumas tarefas, como, por exemplo, navegação pelo site, explicações a respeito das páginas e instruções de preenchimento dos formulários de cadastro de itens. Os itens podem ser buscados pelo nome, organizados por categoria e até filtrados por diferentes etiquetas. As etiquetas são utilizadas para organizar com mais precisão os itens. Com este recurso pode-se separar os itens por marcas, tamanho, série, entre outros.

4.2 Modelagem do sistema

Nesta seção apresentam-se os requisitos funcionais e os requisitos não funcionais do sistema do E-museu.

4.2.1 Requisitos Funcionais e Não Funcionais

Os requisitos funcionais de um sistema determinam as funcionalidades e comportamentos que deverão estar presentes na versão final desse mesmo sistema, já os requisitos não funcionais determinam aspectos como: restrições, segurança, performance e limitações do sistema a ser desenvolvido (ROCHA; MAGALHÃES, 2019). Assim, a seguir são apresentadas duas tabelas com os requisitos do sistema do museu virtual de informática (E-museu). A Tabela 1 possui os requisitos funcionais e a Tabela 2 apresenta os requisitos não funcionais do sistema.

Um aspecto muito importante de todo museu é ser autêntico, ter a confiança dos usuários de que aquilo que está sendo apresentado é verdadeiro. Com essa questão em mente, as funcionalidades do sistema foram projetadas com base na pesquisa de Dutra (2018), a qual estuda e discute como transmitir a sensação de autenticidade do museu virtual ao usuário através da interface do site. Os elementos destacados como indicadores favoráveis a autenticidade do sistema por (DUTRA, 2018) são:

- desenhos de vias que conectam as obras numa galeria de arte;
- áudio guia;
- qualidade de imagens;
- visualização de detalhes com nitidez;
- visualização de obras em ângulos e posições diferentes;
- visualização de outras imagens de contextualização da obra;
- contextualização histórica da obra;
- dados do artista junto com a obra.

Com exceção de áudios guia, imagens de contexto de itens do museu e diferentes ângulos do mesmo item, o restante dos elementos favoráveis estão presentes no sistema web do museu virtual deste projeto. Como elementos que desfavorecem a autenticidade do sistema, citados também por Dutra (2018) tem-se: anúncios publicitários e salas que existem no museu físico, porém não disponíveis no museu virtual. No sistema do E-museu não haverá a presença de anúncios e de salas de um museu físico porque esse não está presente em ambas as Instituições envolvidas no projeto.

Um dos aspectos principais do sistema web é a facilidade de adição de itens novos ao E-museu, tanto pelo administrador, quanto pelos usuários. Sem a necessidade de uma conta no sistema, os usuários poderão sugerir novos itens para o museu virtual, fornecendo informações sobre o mesmo, assim como uma imagem. Cada sugestão de usuário será analisada pelo

Tabela 1 – Requisitos Funcionais.

Identificador	Detalhes
RF001	O sistema web deve listar os aparelhos (equipamentos e itens de informática) do banco de dados em ordem de criação dos mesmos.
RF002	O sistema web deve possibilitar a busca e a filtragem de aparelhos.
RF003	O sistema web deve listar os dados de um aparelho em específico.
RF004	O sistema web deve apresentar os componentes de um aparelho com um link de acesso a página própria deles.
RF005	O sistema web deve apresentar uma assistente virtual em todas as suas telas.
RF006	O sistema web deve disponibilizar opções para interagir com a assistente virtual.
RF007	O sistema web deve apresentar instruções referentes a navegação e funcionalidades do website, se assim for requisitado a assistente virtual.
RF008	O sistema web deve apresentar informações extras sobre um aparelho na página de detalhes de um item.
RF009	O sistema web deve disponibilizar um formulário para o usuário cadastrar um aparelho por conta própria.
RF010	O sistema web, no formulário de cadastro de algum item, irá conferir de antemão se determinado item já está disponível no banco de dados ou não, antes que seja enviado para aprovação do administrador.
RF011	O administrador deve ter acesso a uma página exclusiva a ele, com o fornecimento de um usuário e senha corretos. Nela será disponibilizada uma tabela com os envios de formulários de usuários comuns.
RF012	O administrador deve ter a opção de editar e validar envios de formulários de usuários comuns, para decidir se serão listados no museu virtual ou não.
RF013	O usuário comum deve ter acesso a todas as páginas do museu virtual, sem necessidade de cadastro, com exceção da página exclusiva ao administrador.
RF014	O usuário pode enviar informações adicionais e sugerir novas etiquetas ao sistema.
RF015	O usuário pode visualizar uma imagem do item adicionado, além de aproximá-la para mais detalhes.
RN016	A página do administrador só poderá ser acessada com o fornecimento do nome de usuário e senha corretos.

Tabela 2 – Requisitos Não Funcionais.

Identificador	Detalhes
RN001	O sistema web será desenvolvido usando o <i>framework</i> Laravel.
RN002	O sistema web será desenvolvido utilizando a linguagem de programação PHP, a linguagem de marcação HTML, a linguagem de estilo CSS e a linguagem de programação JavaScript.
RN003	O banco de dados do sistema web será desenvolvido utilizando a linguagem de consultas SQL.
RN004	O sistema web não deve apresentar lentidão em suas listagens e envios de formulários.
RN005	A interface do sistema web deve ser responsiva e dinâmica.
RN006	A interface do sistema web deve ser simples e fácil de se navegar.
RN007	A interface do sistema web deve funcionar corretamente em várias dimensões de telas.

administrador do sistema web que decidirá se aceita ou não a sugestão de adição do novo item.

A assistente virtual serve mais como um recurso visual pelo qual os usuários do website poderão escolher opções variadas, como a filtragem de itens, instruções de uso do sistema, além de navegação para certas páginas do E-museu, com explicações sobre o conteúdo apresentado. O assistente virtual reage a essas opções com base nas informações do banco de dados. A base do desenvolvimento dessa interação é o Javascript. Nele é possível alterar os elementos da tela atual, sem que seja recarregada a página da web, tornando essa interação mais dinâmica.

4.2.2 Banco de Dados

A organização das tabelas do banco de dados foi projetada com base em um artigo escrito por Galvão e Bernardes (2011), que apesar de se tratar de um Museu virtual da Coleção Etnográfica Carlos Estevão de Oliveira e não possuir relação direta com o tema do E-museu, os mesmos princípios foram aplicados para facilitar a navegação, a disposição das informações, a interação do usuário com os itens do museu e a facilidade de busca por aparelhos específicos no sistema. A modelagem do banco de dados foi projetada para ser eficiente e fácil de ser alterada após a criação das tabelas.

Dentro deste contexto, a Figura 6 apresenta as tabelas projetadas para o banco de dados do sistema E-museu. O ponto mais interessante a ser destacado é a tabela "tags". As etiquetas (*tags*) são um recurso para categorizar os itens do museu com mais aprofundamento. Alguns exemplos de etiqueta seriam marca, série, cor, tamanho, entre outros. A tabela "*extras*" armazena todas as informações extras que os aparelhos podem vir a ter ou não. A organização dos aparelhos eletrônicos do museu por categorias, peso, altura, largura, comprimento, marca, série, modelo, código e data foi planejada desta forma para facilitar a filtragem das informações a serem apresentadas ao usuário do E-museu, visto que a exagerada exposição de informações mais atrapalha do que ajuda o usuário. O banco de dados apresenta também uma tabela para contas de administrador e uma tabela de colaboradores: a tabela contendo senha e nome de usuário é reservada apenas aos administradores do sistema, enquanto na tabela dos colaboradores são armazenados o nome completo do colaborador e o seu e-mail. Com o uso da tabela "*locks*" é possível trancar itens sendo modificados por um usuário administrador, ou seja, impedir que outro usuário administrador também modifique o mesmo item. Este recurso não permite que ocorram inconsciências nos dados. Essas tabelas foram geradas através do Laravel que possui ferramentas próprias para isso.

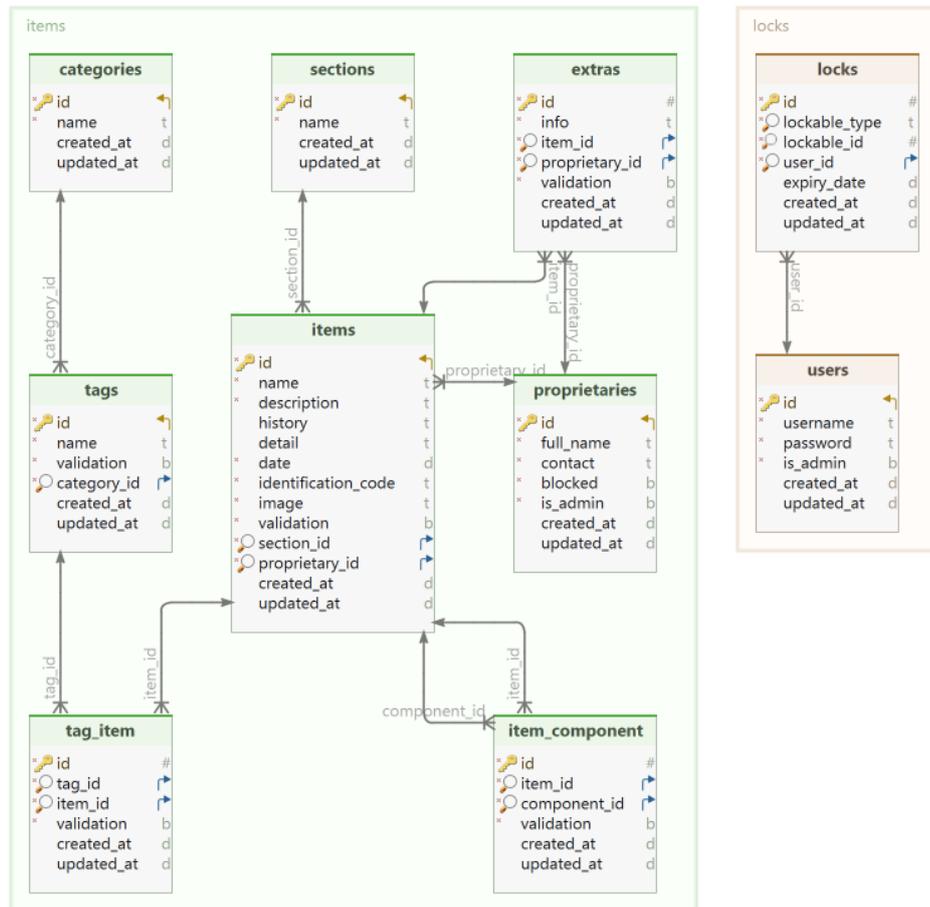


Figura 6 – Modelagem do banco de dados do sistema E-museu com base no trabalho de Galvão e Bernardes (2011).

4.3 Apresentação do sistema

O visual do museu virtual foi projetado para ter uma interface intuitiva, sem excesso de informações e de fácil interação. Desta forma, permite que todo e qualquer usuário consiga navegar pelas páginas do website com facilidade e ao mesmo tempo encontre o que procura. Praticamente todas as telas podem ser acessadas por qualquer usuário sem a necessidade de um cadastro, com exceção da tela de validação de envios dos usuários. Em todas as telas, a assistente virtual está presente para fornecer várias informações, dependendo do contexto e da tela em que o usuário se encontrar. Além disso, perguntas genéricas podem ser realizadas pelo usuário e respondidas pela assistente.

A tela inicial do site, Figura 7, apresenta brevemente o museu virtual, além da disponibilidade da assistente virtual para dar direções de como navegar pelo mesmo.



Figura 7 – Tela inicial do sistema.

Fonte: Autoria própria (2024).

A tela mais importante do museu é a tela de exploração, a qual é apresentada na Figura 8. Nela o usuário pode navegar por categorias de item, filtrar a listagem por etiquetas (por exemplo, marca ou série), visualizar imagens e algumas informações de pré-visualização dos itens disponíveis. Se o usuário desejar, ele pode visualizar mais detalhes de um item específico do museu, clicando sobre ele (Figura 9).

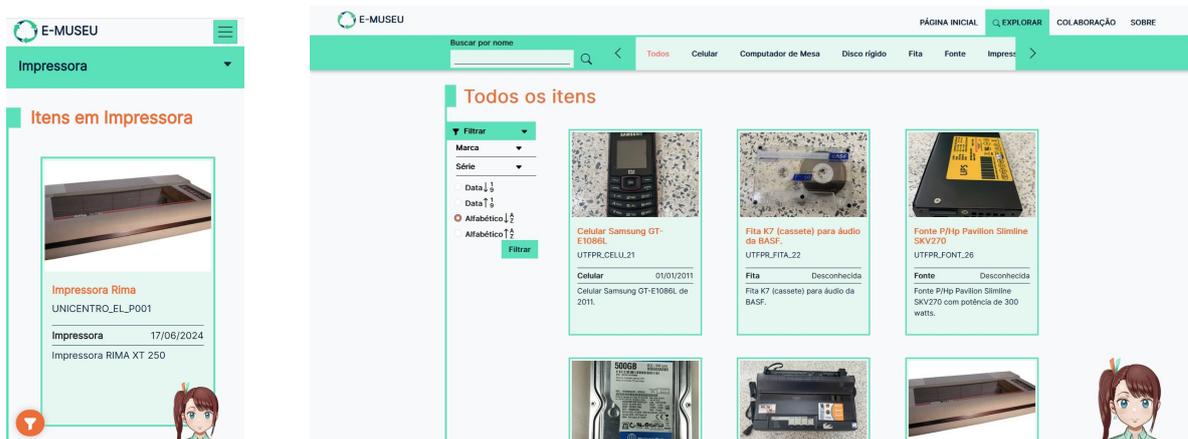


Figura 8 – Tela onde apresentam-se os itens do site que podem ser organizados por categoria.

Fonte: Autoria própria (2024).

A Figura 9 apresenta os detalhes de um item, como a imagem, história, descrição breve, detalhes técnicos, linhas do tempo, data, etiquetas associadas e informações extra. A assistente virtual pode dar detalhes sobre cada seção de informação, caso seja requisitado. Também é possível enviar informações extra sobre o item através da opção acima da imagem do item.

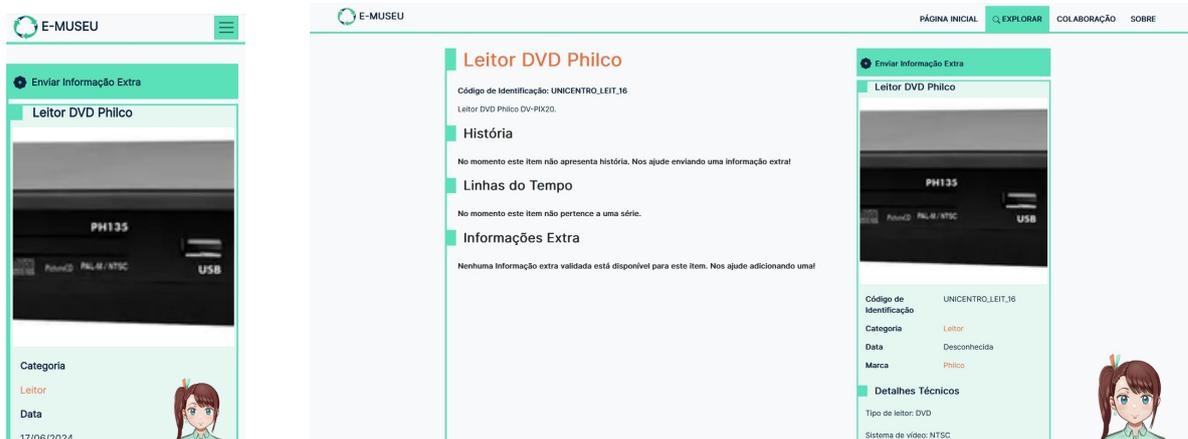


Figura 9 – Tela onde apresentam-se os detalhes, informações, o visual e a linhas do tempo de determinado item.

Fonte: Autoria própria (2024).

Em relação a linha do tempo, a Figura 10 mostra um exemplo em que processadores de uma mesma família (marca) são apresentados.



Figura 10 – Tela que mostra a linha do tempo de processadores.

Fonte: Autoria própria (2024).

A tela da Figura 11 apresenta o formulário para sugestão de itens para o E-museu. Essa página permite cadastrar um item pelo visitante, ou seja, pelo colaborador. Quando um item é cadastrado, o colaborador insere o nome do item, seleciona a categoria a que pertence (p.e., teclado, computador, mouse), informa a data de lançamento, escreve uma breve descrição, detalhes técnicos e história. Além disso, o colaborador pode adicionar uma etiqueta para pesquisa (por exemplo, marca, série, tamanho), informações extras e componentes associados. Como exemplo, pode-se citar o computador que possui memória, disco rígido, entre outros componentes associados. Importante salientar que a tela de visualização de envio dos itens pelos colaboradores, só pode ser acessada pelo administrador do sistema, com o usuário e senha corretos. Nela o administrador poderá modificar e aprovar o item enviado pelo colaborador.

The image shows two screenshots of the E-MUSEU web application. The left screenshot displays the 'Cadastrare um item' form, which includes fields for 'Nome do Item*', 'Categoria*', 'Data de Lançamento*', and 'Descrição Breve*'. The right screenshot shows the 'Colabore com um item' form, which includes fields for 'Nome do Item*', 'Categoria*', 'Data de Lançamento', 'Descrição Breve*', and 'Detalhes Técnicos'. It also features sections for 'Etiquetas', 'Informação Extra', and 'Componentes', each with a '+' icon to add more. A 'Enviar' button is located at the bottom right of the form. Both screenshots include a navigation bar at the top with 'PÁGINA INICIAL', 'EXPLORAR', 'COLABORAÇÃO', and 'SOBRE'.

Figura 11 – Tela onde se disponibiliza o formulário de cadastro de novos itens ao sistema.

Fonte: Autoria própria (2024).

Na tela apresentada pela Figura 12 informações sobre as entidades envolvidas na criação do sistema são apresentadas.

The image shows a screenshot of the E-MUSEU web application's 'Sobre o Museu Virtual' page. The page features a navigation bar at the top with 'PÁGINA INICIAL', 'EXPLORAR', 'COLABORAÇÃO', and 'SOBRE'. The main content area includes a section titled 'Sobre o Museu Virtual' with a paragraph describing the museum's creation and a contact email. Below this is a section for 'Projeto Tecno-lixo' with a paragraph and a 'ACESSAR UTFPR' button. Another section for 'Projeto E-lixo' is partially visible. An image of a computer monitor and keyboard is shown on the right side of the page. A small cartoon character is visible in the bottom right corner.

Figura 12 – Tela sobre as entidades envolvidas na criação do sistema.

Fonte: Autoria própria (2024).

O visual das telas para aparelhos móveis também é apresentado. Os elementos foram reorganizados para facilitar o uso em telas menores e mais estreitas. O uso do Bootstrap, detalhado na Subseção 3.1.4, foi essencial para este processo, pois com o uso de suas classes CSS o site fica dinâmico para se adaptar a diferentes tamanhos de tela.

4.4 Implementação do sistema

Nesta seção, são detalhadas as principais funcionalidades do sistema. Cada funcionalidade será acompanhada por exemplos de código e uma explicação detalhada para melhor compreensão de sua implementação. Os processos foram realizados com base no que a documentação do Laravel¹ apresenta.

4.4.1 Criação do Sistema

O primeiro passo foi instalar, na máquina de desenvolvimento, o PHP e o Composer, o gerenciador de dependências utilizado pelo Laravel. É essencial que ambos estejam instalados antes de prosseguir com a criação do projeto. Utilizando a documentação oficial do Laravel como referência, executou-se o comando de criação de projeto, gerando todos os arquivos necessários para seu funcionamento, incluindo uma página de exemplo.

4.4.2 Configuração do Sistema

A configuração do sistema baseia-se na atualização das variáveis de ambiente, permitindo o início do desenvolvimento. O ponto mais crucial desta etapa é garantir o registro correto das informações necessárias para a conexão com o banco de dados. Além das variáveis de senha e usuário, a variável `DB_CONNECTION` deve ser configurada como `'mysql'`, pois o MySQL foi o sistema de gerenciamento de banco de dados escolhido para o projeto. Como demonstrado na listagem 1, as variáveis de ambiente do banco de dados foram configuradas de acordo com as instruções da documentação Laravel (os valores reais de `DB_USERNAME` e `DB_PASSWORD` não são mostrados, haja vista que são confidenciais).

Listagem 1 – Variáveis de ambiente para a configuração do banco de dados.

```
1 DB_CONNECTION=mysql
2 DB_HOST=127.0.0.1
3 DB_PORT=3306
4 DB_DATABASE=laravel
5 DB_USERNAME=root
6 DB_PASSWORD=
```

Fonte: Autoria própria (2024).

¹ <https://laravel.com/docs/11.x>

4.4.3 Criação das Migrations

Com a configuração do banco de dados realizada, o passo seguinte foi criar as tabelas do banco de dados, com base na modelagem apresentada na subseção 4.2.2. Esse procedimento foi realizado no Laravel através das *Migrations*, que são classes que funcionam como controladores de versão do esquema do banco de dados. Cada *Migrations* representa uma tabela, permitindo que a criação e a remoção das tabelas sejam feitas de forma simultânea com a execução de um único comando.

Utilizando o comando `'php artisan make:migration create_items_table'` do Laravel, um arquivo contendo a estrutura base da *Migration* é criada no projeto. A partir disso, modificações foram realizadas para que as tabelas fossem criadas com as colunas desejadas. No exemplo da listagem 2, tem-se a *Migration* para a tabela de itens do banco de dados. Nela pode-se visualizar que a classe tem duas funções, `up()` e `down()`. Em `up()`, a criação da tabela é configurada a partir da função `Schema::create()`, onde são fornecidas as colunas a serem criadas junto com suas particularidades, como: tipo de dado e se faz papel de chave estrangeira (id utilizado para relacionar um elemento de uma tabela com um elemento de outra tabela). Enquanto a função `down()` tem apenas o intuito de eliminar a tabela do banco de dados.

Vale destacar ainda nesta subseção que a *Migration* para usuários administradores é uma adaptação da *Migration users* que vem por padrão em todo projeto Laravel. Após a criação das *Migrations* e a execução do comando `'php artisan migrate'`, a função `up()` de todas as *Migrations* ainda não realizadas é executada, criando-se assim, as tabelas do banco de dados.

Listagem 2 – Migration da tabela items do banco de dados.

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     public function up(): void
10    {
11        Schema::create('items', function (Blueprint $table) {
12            $table->id();
13            $table->string('name');
14            $table->text('description');
15            $table->text('history')->nullable();
16            $table->text('detail')->nullable();
17            $table->date('date');
18            $table->string('identification_code');
19            $table->string('image');
20            $table->boolean('validation')->default(0);
21            $table->foreignId('section_id')
22                ->constrained()
23                ->onDelete('cascade');
24            $table->foreignId('proprietary_id')
25                ->constrained('proprietaries')
26                ->onDelete('cascade');;
27            $table->timestamps();
28        });
29    }
30
31    public function down(): void
32    {
33        Schema::dropIfExists('items');
34    }
35 };

```

Fonte: A autoria própria (2024).

4.4.4 Implementação de Models

Para interagir com o banco de dados sem a necessidade de escrever códigos SQL, o Laravel oferece um recurso chamado *Eloquent*. O *Eloquent* é um *Object Relational Mapper* (ORM), que tem como função relacionar os dados de uma tabela do banco de dados ao objeto que ele representa. Com este recurso é possível realizar consultas, inserções, atualizações ou remoções de registros de uma tabela no banco de dados com o uso de uma função.

Como exemplo, tem-se o *Model* da tabela *'proprietaries'* na listagem 3. Nela pode-se destacar mais uma das funcionalidades do *Eloquent*, o relacionamento entre models. Ao exe-

cutar a função *items()* do *Model Proprietary*, obtêm-se todos os itens registrados pelo usuário. Essa relação é feita através das chave estrangeira *'proprietary_id'* na tabela *'items'*, como pode ser observado na linha 24 da listagem 2.

Listagem 3 – Model de Proprietary.

```

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 use Illuminate\Database\Eloquent\Relations\MorphMany;
9
10 class Proprietary extends Model
11 {
12     use HasFactory;
13
14     protected $fillable = [
15         'full_name',
16         'contact',
17         'blocked',
18         'is_admin'
19     ];
20
21     protected $table = 'proprietaries';
22
23     public function items() {
24         return $this->hasMany(Item::class);
25     }
26
27     public function locks(): MorphMany
28     {
29         return $this->morphMany(Lock::class, 'lockable');
30     }
31 }

```

Fonte: Autoria própria (2024).

4.4.5 Implementação de Factories

Para testar o banco de dados e as páginas em que são listados os elementos, *Factories* exercem um papel fundamental. O propósito de uma *Factory* é gerar, de acordo com as configurações feitas, registros no banco de dados de forma aleatória. No caso deste projeto, foi utilizado a biblioteca *Faker* do PHP², que gera aleatoriamente diferentes tipos de valores, dependendo de sua configuração. Na listagem 4 apresenta-se um exemplo de uma *Factory*. Nela observa-se

² <https://fakerphp.org>

as diferentes formas de utilizar o *Faker*, como gerar nomes, parágrafos de texto, números e até caminhos para imagens.

Listagem 4 – Factory de itens.

```

1 <?php
2
3 namespace Database\Factories;
4
5 use Illuminate\Database\Eloquent\Factories\Factory;
6 use App\Models\Proprietary;
7 use App\Models\Brand;
8 use App\Models\Class;
9 use App\Models\Section;
10
11 class ItemFactory extends Factory
12 {
13     public function definition(): array
14     {
15         return [
16             'name' => $this->faker->unique()->sentence ,
17             'description' => $this->faker->paragraph ,
18             'history' => $this->faker->paragraph(500),
19             'detail' => $this->faker->text ,
20             'date' => $this->faker->date ,
21             'identification_code' => $this
22                 ->faker
23                 ->unique ()
24                 ->numberBetween(1, 1000),
25             'image' => $this->faker->imageUrl(500, 500),
26             'validation' => $this->faker->boolean ,
27             'section_id' => Section::pluck('id')->random() ,
28             'proprietary_id' => Proprietary::pluck('id')->random() ,
29         ];
30     }
31 }

```

Fonte: Autoria própria (2024).

4.4.6 Implementação de *Seeders*

A classe *Seeder* exerce a função de inserir no banco de dados, registros essenciais para o funcionamento do sistema. Mas, também pode ser utilizado para testes, se assim for programado. No caso deste projeto, havia a necessidade de categorias de item pré-determinadas no banco de dados para que algumas funcionalidades fossem testadas. Na listagem 5 foi disponibilizado um exemplo de um *Seeder* para a tabela de categoria de itens, nomeado no banco de dados como *'section'*. Pode-se analisar a partir do exemplo que a classe *Seeder* utiliza a classe *Model*, detalhada na subseção 4.4.4, da qual se deseja criar registros.

Listagem 5 – Seeder de Section.

```

1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7 use App\Models\Section;
8
9 class SectionSeeder extends Seeder
10 {
11     public function run(): void
12     {
13         Section::create([ 'name' => 'Monitor' ]);
14         Section::create([ 'name' => 'Notebook' ]);
15         Section::create([ 'name' => 'Computador de Mesa' ]);
16         Section::create([ 'name' => 'Fone' ]);
17         Section::create([ 'name' => 'Mouse' ]);
18         Section::create([ 'name' => 'Teclado' ]);
19         Section::create([ 'name' => 'Impressora' ]);
20         Section::create([ 'name' => 'Armazenamento' ]);
21         Section::create([ 'name' => 'Placa de Vídeo' ]);
22         Section::create([ 'name' => 'Webcam' ]);
23         Section::create([ 'name' => 'Memória Ram' ]);
24         Section::create([ 'name' => 'Roteador' ]);
25         Section::create([ 'name' => 'Tablet' ]);
26         Section::create([ 'name' => 'Celular' ]);
27         Section::create([ 'name' => 'Placa-mãe' ]);
28         Section::create([ 'name' => 'Processador' ]);
29     }
30 }

```

Fonte: Autoria própria (2024).

4.4.7 Implementação de Views

As *Views* tem ao propósito de apresentar visualmente os dados do sistema ao usuário e geralmente não contam com regras de negócio, já que essa função pertence aos *Controllers*. Elementos como textos, gráficos, imagens e formulários de preenchimento fazem parte desta camada. Apesar da estilização ter sido implementada após a definição dos controllers da aplicação, a estrutura HTML e o CSS (responsável pela estilização de um site) do sistema serão explicados simultaneamente, dado que, a ordem desses fatores não impactem na compreensão da subseção.

Views em Laravel são implementados em extensão Blade. Basicamente são arquivos PHP que também aceitam *tags* de HTML, permitindo a apresentação e manipulação dinâmica de dados em um mesmo arquivo. Funções como *if* (usado como condicional) e *foreach* (usado

para percorrer listas de elementos) podem ser utilizados, tornando extremamente prático o processo de apresentação de dados enviados do sistema pelo *Controller* ao usuário.

Na listagem 6, tem-se um fragmento de uma view em Laravel Blade. Com base no exemplo dado, pode-se analisar algumas das funcionalidades apresentadas nesta subseção. Com uso das diretivas *@if* e *@else*, pode-se abrir e fechar uma condicional diretamente no arquivo que também terá *tags* HTML. Neste caso em específico, a variável *'session'* foi verificada para determinar se há uma mensagem de sucesso presente. Caso a encontre, o conteúdo de dentro da condicional é apresentado, caso contrário, é ignorado. Para se ter acesso às informações enviadas pelo *Controller*, a variável deve ser envolvida por chaves duplas. Ainda na listagem 6, pode-se observar as diretivas *foreach* e *endForeach*, usadas para abrir e fechar um percorrimto de listas. No exemplo dado, a diretiva percorre a variável *'\$errors'* e apresenta as informações usando *'\$error'*. A variável *'\$errors'* armazena erros ocorridos na validação de registro enviados pelo usuário em um formulário.

Juntamente com as diretivas do Laravel Blade, tem-se uma pequena amostra de usos de classes do Bootstrap. Ao adicionar essas classes em *tags* HTML, pode-se personalizar facilmente o visual e a responsividade do site. As classes utilizadas na listagem são: *fw-bold* (deixa o texto em negrito), *m-1* (adiciona margem de *0.25rem*), *p-1* (adiciona preenchimento de *0.25rem*) e *mb-4* (adiciona uma margem de *0.25rem* apenas para a parte de baixo do elemento). Ainda tem-se o uso da biblioteca de ícones nas *tags* *'<i>'*.

Para implementar a interação com a assistente foi utilizado o Javascript. Cada linha do diálogo é organizado em nós e, como mostrado na listagem 7, cada nó possui um id, o texto de diálogo e escolhas para o usuário. As escolhas são de dois tipos, um tipo redireciona para outras páginas do site e o outro avança para outro nó. Para este sistema funcionar, foi necessário criar um interpretador de nós básico, em que se apresenta em tela o conteúdo de *'text'* e as escolhas em forma de botões. Ao selecionar uma escolha, o interpretador verifica o conteúdo de *'choices'* e *'nextId'*. Assim, avança para o respectivo id do nó, enquanto a *Uniform Resource Locator* (URL) direciona o usuário à URL providenciada. O visual da assistente visual foi realizada no Clip Studio Paint, citado na subseção 3.1.9.

Listagem 6 – Fragmento de uma View em Laravel Blade.

```

1 @if (session('success'))
2     <div class="success-div text-wrap fw-bold m-1 p-1">
3         {{ session('success') }}
4     </div>
5 @endif
6
7 @foreach ($errors->all() as $error)
8     <p class="error-div text-wrap fw-bold m-1 mb-4 p-1">
9         <i class="bi bi-exclamation-circle-fill mx-1 h5"></i>
10        {{ $error }}
11    </p>
12 @endforeach

```

Fonte: Autoria própria (2024).

Listagem 7 – Fragmento de diálogo da assistente.

```

1 {
2     id: 2,
3     text: "O que gostaria de saber?",
4     choices: [
5         { text: "Saber mais sobre a página atual", nextId: 3 },
6         { text: "Explorar itens do museu >", url: "items" },
7         { text: "Contribuir com o museu >", url: "items/create" },
8         { text: "Saber mais sobre o museu >", url: "about" },
9         { text: "Entrar em contato com o museu >", nextId: 4 },
10    ]
11 }

```

Fonte: Autoria própria (2024).

4.4.8 Implementação de Controllers

Os *Controllers* são a última camada do padrão MVC, explicado na subseção 3.1.1, sendo responsáveis por mediar a execução das regras de negócio com os dados vindos das camadas *View* e *Model*. Operações como resgatar e inserir dados no banco de dados, validar registros enviados pela *View* e até bloquear acesso e certas ações de usuários são responsabilidade dos *Controllers*.

A partir da listagem 8, apresenta-se uma análise do funcionamento de um *Controller* em Laravel. A função *store*, neste caso, recebe um argumento de tipo *FormRequest*. *FormRequests* são classes PHP utilizadas para validar os dados enviados pelo usuário a partir dos formulários HTML. O Laravel facilita bastante o uso, pois, tendo criado e configurado o arquivo, com as validações necessárias para o caso específico, a execução da classe ocorre de forma automática ao defini-la como argumento da função. Caso a validação não aprove os dados, uma mensagem de erro é enviada para a *View* e apresentada ao usuário. Após a validação, a variável *\$data* recebe os dados validados a partir da função *\$request->validated()*, que resgata os campos validados vindos da requisição. Com esses dados, uma categoria é criada a partir

da função `create` do *Eloquent* na linha 4. E, por fim, o usuário, neste caso o administrador, é redirecionado para a *View 'show'* de uma categoria, mostrando a categoria recém criada com uma mensagem de sucesso.

Para a autenticação do site, foi utilizado o pacote oficial do Laravel chamado *laravel/ui*, instalado via Composer. O pacote possui *Controllers* de autenticação e componentes *front-end* já implementados, possuindo recursos como: autenticação, criação de contas, alteração de senha e confirmação de e-mail. Para que funcionasse corretamente foi necessário apenas realizar algumas configurações e pequenas adaptações, como: remover rotas de criação de contas, alteração de senha e confirmação de e-mail. Essas alterações foram feitas, pois, o sistema de autenticação é direcionado apenas aos administradores do site. Além de tudo, o sistema contém proteção *Cross-site Request Forgery (CSRF)*, que impede requisições forjadas e hashing de senhas.

Listagem 8 – Função `store` de `AdminController`.

```

1 public function store(CategoryRequest $request)
2 {
3     $data = $request->validated();
4     $category = Category::create($data);
5
6     return redirect()
7         ->route('admin.categories.show', $category)
8         ->with('success', 'Categoria adicionada com sucesso. ');
9 }

```

Fonte: Autoria própria (2024).

4.4.9 Implementação de Rotas

Como último ponto a ser detalhado na implementação do website, apresenta-se o exemplo da listagem 9. Rotas são configuradas facilmente no arquivo `web.php` do Laravel. No exemplo destacado, a rota com requisição `get` vai para a página de detalhes de um item. Utilizando a função `'Route::get()'`, definiu-se que a rota será de requisição `get`, e, dos argumentos passados `'/items/{id}'` e `'[itemController::class, 'show']'`. No primeiro argumento passado, definiu-se a qual rota esta função responderá (`'id'` sendo o id do item a ser mostrado), e no segundo argumento definiu-se o *Controller* e a função que responderá por esta rota. Conseguiu-se também atribuir um nome para a rota com o uso de `'->name()'`. Este recurso serve apenas para facilitar a menção a esta rota em outras funções. Finalmente, destaca-se `'->middleware()'`, que funciona como um filtro para requisições. Com o uso deste recurso do Laravel, pode-se facilmente aplicar funções adicionais ao site. No exemplo dado, utilizou-se um *Middleware* customizado de argumento, o `'validate.item'`, que verifica se o item com o id providenciado na rota está validado pelo administrador ou não. Caso o item não esteja validado, o acesso não é garantido ao usuário.

Listagem 9 – Função store de AdminCategoryController.

```

1 Route::get('/items/{id}', [ItemController::class, 'show'])
2     ->name('items.show')
3     ->middleware('validate.item');
```

Fonte: Autoria própria (2024).

4.5 Implantação do E-museu

O E-museu foi implantado em uma máquina virtual cedida pela UTFPR-GP com a seguinte configuração: 2GB de memória RAM, 50GB de espaço em disco, Sistema Operacional Linux Ubuntu³ e servidor web NGINX⁴. O acesso a máquina virtual se dá através do protocolo SSH - *Secure Shell*. Para a implantação do website do E-museu consultou-se a documentação⁵ de implantação de sites Laravel e executou-se as configurações do PHP, do banco de dados MySQL, das variáveis de ambiente, de instalação de dependências do projeto e execução das Migrations. O domínio definido para o website do E-museu ficou e-museu.gp.utfpr.edu.br. A Unicentro se prontificou a hospedar o website do E-museu; contudo, optou-se por mantê-lo na máquina virtual disponibilizada pela UTFPR-GP. Desta forma, é possível gerenciar o servidor NGINX, realizar as manutenções nos códigos e realizar backups periodicamente, uma vez que o sistema web do E-museu será mantido pela e gerenciado na UTFPR.

4.6 Teste de usabilidade do E-museu

O *System Usability Scale* (SUS), elaborado por John Brooke, foi utilizado como método de avaliação da usabilidade do site. Este método é baseado em um questionário que pode ser visualizado no Apêndice A, e um cálculo realizado a partir dos resultados para se obter uma pontuação que simbolize a satisfação dos usuários com o sistema testado. Este cálculo é realizado subtraindo o valor 1 de todas as escalas em questões de número ímpar e subtraindo o valor 5 de todas as escalas em questões de número par. Após isso, soma-se todos os valores obtidos e multiplica-se por 2,5. A partir deste cálculo, obtem-se um número entre 0 e 100, sendo que, 100 simboliza uma usabilidade perfeita e 0 uma usabilidade nada satisfatória (FORÇAN *et al.*, 2021).

Para realizar o teste de usabilidade do E-museu foram selecionados 5 acadêmicos da Unicentro e UTFPR. O questionário, acima mencionado, com 10 perguntas sobre a usabilidade do site foi elaborado no Google Forms⁶ e aplicado para esses alunos. Cada pergunta do teste utilizou a escala de Likert apresentada por (FORÇAN *et al.*, 2021).

³ <https://ubuntu.com>

⁴ <https://nginx.org/en/>

⁵ <https://laravel.com/docs/11.x>

⁶ <https://workspace.google.com/intl/pt-BR/lp/forms/>

A Tabela 3 apresenta os resultados obtidos por meio do *feedback* coletado depois da aplicação do formulário. A primeira coluna contém os cinco usuários que testaram o site, e nas demais colunas temos as respostas dadas para cada questão, em que 1 simboliza a discordância total e 5, a concordância total.

Tabela 3 – Resultados do teste de Usabilidade.

Usuários	Q. 1	Q. 2	Q. 3	Q. 4	Q. 5	Q. 6	Q. 7	Q. 8	Q. 9	Q. 10
Usuário 1	3	1	5	1	4	2	5	1	4	1
Usuário 2	3	1	5	1	4	1	4	1	5	1
Usuário 3	3	1	5	1	4	1	4	1	3	1
Usuário 4	5	1	5	1	5	3	5	1	5	1
Usuário 5	5	1	5	1	5	1	5	1	5	1

Após coletar as respostas do questionário, foi necessário realizar o cálculo do método de avaliação SUS, detalhado anteriormente. Para cada amostra, dos usuários 1 ao 5, as pontuações foram as seguintes:

- Usuário 1: 87,5;
- Usuário 2: 87,5;
- Usuário 3: 85,0;
- Usuário 4: 95,0;
- Usuário 5: 100,0;

Realizando uma média aritmética das cinco amostras (onde a soma dos valores é dividida pela quantidade de valores), obteve-se a pontuação 91, que com base no método SUS, é uma pontuação excelente.

5 CONSIDERAÇÕES FINAIS

Os museus apresentam boa parte da história e da cultura de um povo e não precisam estar em um local físico. Com isso, surgem os museus virtuais em que as pessoas podem através da Internet visitar o acervo disponível. Sendo assim, o objetivo deste trabalho foi desenvolver um sistema web para um museu virtual de informática com o propósito de registrar e apresentar a história de computadores, peças de computadores ou itens de informática. A ideia inicial surgiu porque a Universidade Tecnológica Federal do Paraná (UTFPR) e a Universidade Estadual do Centro-Oeste (Unicentro) possuem projetos ligados ao lixo eletrônico. A UTFPR possui o projeto Tecno-Lixo: oficina do aprender (HAYNE, 2022) e a Unicentro o projeto E-Lixo (Ré; THOMEN, 2021). Os projetos recebem peças de computadores e muitas delas por serem bem antigas e diferentes despertaram o interesse em guardá-las, pesquisar sobre sua história e mostrar para outras pessoas o quão interessante elas são.

Desta forma, este trabalho mostrou a implementação do Museu virtual de informática, chamado de E-museu, que apresenta e resgata um pouco da história de peças ou itens de informática que já foram utilizados, e que hoje em dia podem estar caindo no esquecimento. Assim, o E-museu armazena os detalhes da aparência, especificações, usos e outros aspectos das peças. Ainda, nesta mesma plataforma, as peças são expostas a qualquer pessoa com acesso a Internet, sem necessidade de criar contas ou agendar horários para a sua visita.

O desenvolvimento do trabalho exigiu o conhecimento e a aplicação de vários conceitos e tecnologias trabalhados durante o Curso de Tecnologia em Sistemas para Internet da UTFPR - Campus Guarapuava. Foram colocados em prática os princípios e fundamentos de programação de computadores, design gráfico, interação homem-computador, banco de dados, além de princípios de análise e projeto de software, entre outros. Isso para planejar o visual e o fluxo de funcionamento do sistema, bem como para que o banco de dados fosse o mais eficiente e otimizado possível. As tecnologias utilizadas durante o curso, como HTML, Javascript, PHP, CSS e SQL foram aplicadas no desenvolvimento do sistema do E-museu. A documentação e o planejamento de como resolver problemas e definir os estágios da execução desse projeto foram baseados no aprendizado obtido durante o curso.

Durante o desenvolvimento do trabalho foram enfrentados alguns desafios, os quais foram superados. Dentre eles, destacam-se:

1. **Falta de armazenamento para imagens dos itens:** inicialmente, havia a intenção de armazenar diversas imagens para cada item. No entanto, para economizar espaço de armazenamento, decidiu-se por armazenar apenas uma imagem por item.
2. **Local de implantação do sistema:** no início, não havia um local para a disponibilização do sistema do E-museu na Internet sem a necessidade de pagamento. Contudo, a UTFPR cedeu uma máquina virtual em que o servidor web NGINX foi instalado. Assim,

o servidor cedido pela UTFPR supriu adequadamente as necessidades de implantação do sistema.

3. **Método de categorização dos itens:** como os itens são muito diferentes e com características variadas, o desafio era fazer uma categorização da forma mais eficaz possível. Assim, foram utilizadas etiquetas para complementar a organização dos itens por categoria, facilitando assim a gestão e a busca pelos itens.
4. **Definição do tamanho da amostra para o teste de usabilidade:** como o E-museu foi disponibilizado na Internet, a questão era como se ter um tamanho de amostra que representa-se todo o público da web. Nesse caso, foi difícil mensurar, e; ao final, o tamanho da amostra de usuários para o teste de usabilidade ficou reduzido, o que pode ter impactado a abrangência dos resultados obtidos.

Por fim, o sistema web do E-museu, disponível em e-museu.gp.utfpr.edu.br, apresentou um visual amigável, responsivo e intuitivo o que foi confirmado a partir das pontuações obtidas no teste de usabilidade que atingiu a pontuação de 91, que o classifica como excelente. Além disso, a partir do *feedback* recebido, das pontuações obtidas no teste de usabilidade, da análise dos requisitos funcionais e não funcionais e dos objetivos específicos alcançados, é possível concluir que o objetivo geral foi atingido. Como efeito, espera-se que o E-museu desperte o interesse dos acadêmicos e alunos da região para que conheçam os cursos das Universidades envolvidas no desenvolvimento e implantação do sistema. Além disso, como se trata de um museu virtual de informática que também conta com a colaboração dos usuários para a construção de seu acervo virtual, o E-museu não apresenta uma grande quantidade inicial de itens. Espera-se que a quantidade aumente ao longo do tempo com a cooperação dos usuários que enviarem sugestões ao website.

5.1 Trabalhos Futuros

Como melhorias no sistema web do E-museu, sugere-se a:

- geração de relatórios para impressão pelo administrador dos itens mais pesquisados e mais cadastrados, bem como das páginas mais acessadas;
- criação de páginas para cadastro de artigos, relacionados a história da computação e dos itens de informática, escritos por alunos e professores das duas Instituições envolvidas;
- associação de cada item a links, quando possível, que redirecionam o usuário a vídeos, artigos e outras informações relacionadas ao item;

- adição de verificação Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) para evitar cadastro de itens e envio de informações extras por bots (softwares usados para simular ações humanas);
- atualização contínua da interface para manter o sistema dentro das tendências de User Interface (UI);
- implementação de testes automatizados de interface;
- utilização do recurso de internacionalização do Laravel (*lang*) para facilitar a tradução do website para outras línguas;
- distribuição das funções dos Controllers entre *Services* e *Actions*.

Por fim, o trabalho terá continuidade com um novo projeto de extensão chamado E-museu: Museu virtual de informática, o qual será vinculado ao projeto Tecno-lixo: oficina do aprender da UTFPR-GP e E-lixo da Unicentro.

REFERÊNCIAS

- AMAZONWEBSERVICESINC. **O que é AJAX?** 2024. Disponível em: <https://aws.amazon.com/pt/what-is/ajax/>. Acesso em: 18 de Maio de 2024.
- BAUER, J. **MUSEU, MUSEOLOGIA E MUSEOGRAFIA**. 2022. Disponível em: <https://www.triscele.com.br/triscele/museu-museologia-e-museografia#:~:text=Surge%20o%20Primeiro%20Museu%3A%20a,em%20templos%2C%20santu%C3%A1rios%20e%20tumbas>. Acesso em: 03 de outubro de 2022.
- BOOTSTRAPTEAM. **Get started with Bootstrap**. 2024. Disponível em: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>. Acesso em: 18 de Maio de 2024.
- CELSYSINC. **Clip Studio**. 2024. Disponível em: <https://www.clipstudio.net/pt/>. Acesso em: 18 de Maio de 2024.
- DUTRA, L. F. **GESTÃO DA INFORMAÇÃO E TECNOLOGIAS: DIRETRIZES PARA PROJETOS DA INTERFACE DE MUSEUS VIRTUAIS NO ÂMBITO DA AUTENTICIDADE**. 2018. Disponível em: https://repositorio.ufmg.br/bitstream/1843/ECIP-B88MAM/1/dissertacao_de_mestrado_ppg_goc_larissa_fernandes_dutra.pdf. Acesso em: 26 de Novembro de 2022.
- EICHLER, M. L.; PINO, J. C. D. **Museus virtuais de ciências: uma revisão e indicações técnicas para o projeto de exposições virtuais**. 2007. Disponível em: <https://seer.ufrgs.br/index.php/renote/article/view/14377/8274>.
- FIGMA. **Figma**. 2024. Disponível em: <https://www.figma.com/pt-br/about/>. Acesso em: 18 de Maio de 2024.
- FORÇAN, L. R. *et al.* **System Usability Scale Based on Paraconsistent Annotated Evidential Logic ET**. 2021. Disponível em: <https://rsdjournal.org/index.php/rsd/article/view/17617/15786>. Acesso em: 25 de Junho de 2024.
- GALVÃO, G. K. A.; BERNARDES, D. A. de M. **A organização da informação como instrumento de preservação e acesso ao Museu Virtual da Coleção Etnográfica Carlos Estevão de Oliveira**. 2011. Disponível em: <http://200.156.20.26/index.php/ppgpmus/article/view/119/170>.
- GITHUBINC. **About GitHub and Git**. 2024. Disponível em: <https://docs.github.com/en/get-started/start-your-journey/about-github-and-git>. Acesso em: 18 de Maio de 2024.
- GUEDES, M. **O que é MVC?** 2023. Disponível em: [https://www.treinaweb.com.br/blog/o-que-e-mvc#:~:text=O%20MVC%20sugere%20uma%20maneira,camada%20de%20controle%20\(controller\)](https://www.treinaweb.com.br/blog/o-que-e-mvc#:~:text=O%20MVC%20sugere%20uma%20maneira,camada%20de%20controle%20(controller)). Acesso em: 18 de Maio de 2024.
- HANASHIRO, A. **VS Code - O que é e por que você deve usar?** 2021. Disponível em: <https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar>. Acesso em: 18 de Maio de 2024.
- HAYNE, D. R. *et al.* Tecno-lixo: Oficina do aprender. **XII Seminário de Extensão e Inovação XXVII Seminário de Iniciação Científica e Tecnológica da UTFPR, Santa Helena, 2022**. Disponível em: <https://www.even3.com.br/anai/seisicite2022/>.

HENRIQUES, R. **Os museus virtuais: conceito e configurações**. 2018. Disponível em: <https://recil.ensinulusofona.pt/bitstream/10437/9299/1/6337-49-19613-1-10-20181218.pdf>. Acesso em: 26 de Novembro de 2022.

L., A. **O Que é GitHub, Para Que Serve e Como Usar**. 2024. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-github>. Acesso em: 18 de Maio de 2024.

LARAGON. **Laragon provides you a modern powerful development environment that countless people love to use every day**. 2024. Disponível em: <https://laragon.org/about/>. Acesso em: 18 de Maio de 2024.

LARAVELHOLDINGSINC. **Laravel**. 2024. Disponível em: <https://laravel.com>. Acesso em: 18 de Maio de 2024.

LIMA, G. **Bootstrap: O que é, Documentação, como e quando usar**. 2023. Disponível em: <https://www.alura.com.br/artigos/bootstrap>. Acesso em: 18 de Maio de 2024.

MELO, D. **O que é Laravel? [Guia para iniciantes]**. 2021. Disponível em: <https://tecnoblog.net/responde/o-que-e-laravel-guia-para-iniciantes/>. Acesso em: 18 de Maio de 2024.

NATAL, G. **Levantamento de requisitos no desenvolvimento ágil de software**. 2018. Disponível em: https://dadospdf.com/download/levantamento-de-requisitos-no-desenvolvimento-agil-de-software_5abb7777b7d7bc0142f74a14_.pdf. Acesso em: 18 de Maio de 2024.

PESSOA, A. B. **Laragon: Vantagens e desvantagens**. 2023. Disponível em: <https://community.revelo.com.br/laragon-vantagens-e-desvantagens/>. Acesso em: 18 de Maio de 2024.

REDHATINC. **O que é IDE (Ambiente de desenvolvimento integrado)?** 2023. Disponível em: <https://www.redhat.com/pt-br/topics/middleware/what-is-ide>. Acesso em: 18 de Maio de 2024.

ROCHA, R. da S.; MAGALHÃES, T. M. de. **ENGENHARIA DE REQUISITOS**. 2019. Disponível em: <https://www.fsd.edu.br/wp-content/uploads/2019/12/artigo27-RAFAEL-e-TERESINHA.pdf>. Acesso em: 20 de Novembro de 2022.

Ré, A. M. D.; THOMEN, M. A. F. **Cartilha de sugestões para atividades em oficinas pedagógicas utilizando de informática LIXOELETRÔ**. 2021. Disponível em: <https://www3.unicentro.br/decomp/wp-content/uploads/sites/77/2021/09/CartilhaLixoEletronico.pdf>. Acesso em: 25 de setembro de 2022.

SCHWEIBENZ, W. **The “Virtual Museum”¹ : New Perspectives For Museums to Present Objects and Information Using the Internet as a Knowledge Base and Communication System**. 1998. Disponível em: http://www.informationswissenschaft.org/wp-content/uploads/isi/isi1998/14_isi-98-dv-schweibenz-saarbruecken.pdf. Acesso em: 26 de Novembro de 2022.

SILVA, H. **81% da população brasileira acessou a internet em 2021, diz pesquisa; TV supera computador como meio**. 2022. Disponível em: <https://g1.globo.com/tecnologia/noticia/2022/06/21/81percent-da-populacao-brasileira-acessou-a-internet-em-2021-diz-pesquisa.ghtml>. Acesso em: 25 de setembro de 2022.

SOUZA, I. de. **Entenda o que é jQuery, para que serve e como usar no seu site**. 2021. Disponível em: <https://rockcontent.com/br/blog/jquery/>. Acesso em: 18 de Maio de 2024.

VOLTOLINI, R. **Museu da Informática é inaugurado em SP; faça agora um tour online!** 2015. Disponível em: <https://www.tecmundo.com.br/brasil/85078-museu-informatica-inaugurado-sp-faca-tour-online.htm>. Acesso em: 03 de outubro de 2022.

WIKIPEDIA. **Wikipedia:About**. 2024. Disponível em: <https://en.wikipedia.org/wiki/Wikipedia:About>. Acesso em: 13 de Maio de 2024.

WISECODERSGMBH. **DBSCHEMA DOCUMENTATION**. 2024. Disponível em: <https://dbschema.com/documentation/>. Acesso em: 18 de Maio de 2024.

WISECODERSGMBH. **WISE CODERS GMBH**. 2024. Disponível em: <https://dbschema.com/company.html>. Acesso em: 18 de Maio de 2024.

APÊNDICE A – Questionário do SUS (System Usability Scale)

TESTE DE USABILIDADE**1. Eu usaria este sistema com frequência.**

- Discordo totalmente Discordo parcialmente Não concordo nem discordo
 Concordo parcialmente Concordo totalmente

2. Esse sistema é desnecessariamente complexo.

- Discordo totalmente Discordo parcialmente Não concordo nem discordo
 Concordo parcialmente Concordo totalmente

3. O sistema é fácil de usar.

- Discordo totalmente Discordo parcialmente Não concordo nem discordo
 Concordo parcialmente Concordo totalmente

4. Eu precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.

- Discordo totalmente Discordo parcialmente Não concordo nem discordo
 Concordo parcialmente Concordo totalmente

5. As funções do sistema são muito bem integradas.

- Discordo totalmente Discordo parcialmente Não concordo nem discordo
 Concordo parcialmente Concordo totalmente

6. O sistema apresenta muita inconsistência.

- Discordo totalmente Discordo parcialmente Não concordo nem discordo
 Concordo parcialmente Concordo totalmente

7. As pessoas aprenderão como usar esse sistema rapidamente.

- Discordo totalmente Discordo parcialmente Não concordo nem discordo
 Concordo parcialmente Concordo totalmente

8. O sistema é confuso para usar.

- Discordo totalmente Discordo parcialmente Não concordo nem discordo
 Concordo parcialmente Concordo totalmente

9. Senti confiança ao utilizar o sistema.

- Discordo totalmente Discordo parcialmente Não concordo nem discordo
 Concordo parcialmente Concordo totalmente

10. **Precisei aprender várias coisas novas antes de conseguir utilizar o sistema.**

Discordo totalmente Discordo parcialmente Não concordo nem discordo

Concordo parcialmente Concordo totalmente