

UNIVERSIDADE PAULISTA

GUSTAVO HENRIQUE SANSIVIERI DE VECCHIO CUEBRA - T923FG2

ISAC FERREIRA DE SOUZA - G750381

LEANDRO MORAES DE SOUZA - G7862D1

THAIS REIS DA SILVA - T917AG6

PROJETO DE TOTEM INTERATIVO PARA MUSEU MULTITEMÁTICO
“O HOMEM NA LUA”

SOROCABA

2024

GUSTAVO HENRIQUE SANSIVIERI DE VECCHIO CUEBRA - T923FG2

ISAC FERREIRA DE SOUZA - G750381

LEANDRO MORAES DE SOUZA - G7862D1

THAIS REIS DA SILVA - T917AG6

Projeto Integrado Multidisciplinar (PIM) para o curso de Análise e Desenvolvimento de Sistemas apresentado ao Centro Universitário Paulista – UNIP de Sorocaba, como forma de avaliação parcial.

Orientador: Prof. Esp. Reverdan Almeida Sparinger

SOROCABA

2024

GUSTAVO HENRIQUE SANSIVIERI DE VECCHIO CUEBRA - T923FG2

ISAC FERREIRA DE SOUZA - G750381

LEANDRO MORAES DE SOUZA - G7862D1

THAIS REIS DA SILVA - T917AG6

PROJETO DE TOTEM INTERATIVO PARA MUSEU MULTITEMÁTICO

“O HOMEM NA LUA”

Projeto Integrado Multidisciplinar (PIM) para o curso de Análise e Desenvolvimento de Sistemas apresentado ao Centro Universitário Paulista – UNIP de Sorocaba, como forma de avaliação parcial.

Aprovado em:

BANCA EXAMINADORA

Prof. Esp. Alex Sampaio

Universidade Paulista – UNIP

_____/____/____

Prof. Esp. Reverdan Almeida Springer

Universidade Paulista – UNIP

_____/____/____

Prof. Me. Waldir Silva

Universidade Paulista – UNIP

_____/____/____

SOROCABA

2024

RESUMO

O projeto de desenvolvimento de um software para um totem destinado a um museu multitemático para uma exposição com o tema: "O Homem na Lua", foi solicitado por uma instituição sem fins lucrativos, visando promover o acesso à informação e a interação dos visitantes com o acervo do museu. O desenvolvimento de um software envolve a análise dos usuários, concepção, planejamento e implementação de funcionalidades que atendam a essas demandas, garantindo a eficácia e usabilidade do produto, através de melhorias e manutenção, possibilitando para o usuário uma experiência, intuitiva, envolvente, satisfatória e de fácil acesso, integrando a tecnologia interativa em um contexto cultural evidenciando o potencial transformador da tecnologia no fortalecimento da conexão entre o público e o patrimônio cultural.

Palavras-Chave: Tecnologia, Software, Totem, Museu

ABSTRACT

The project to develop software for a totem for a multi-themed museum for an exhibition with the theme: "The Man on the Moon", was requested by a non-profit institution, aiming to promote access to information and interaction of visitors with the museum's collection. Software development involves user analysis, design, planning and implementation of features that meet these demands, ensuring the effectiveness and usability of the product, through improvements and maintenance, providing the user with an intuitive, engaging, satisfactory experience. and easily accessible, integrating interactive technology into a cultural context, highlighting the transformative potential of technology in strengthening the connection between the public and cultural heritage.

Keywords: technology, software, totem, museum.

LISTA DE ILUSTRAÇÕES

Figura 1: Termo de Abertura de Projeto.....	15
Figura 2: Princípios da qualidade de software da norma ISO 9126	20
Figura 3: Infraestrutura de redes.....	27
Figura 4 - Mapa do museu com infraestrutura de redes.....	28
Figura 5 - Legenda de redes	28
Figura 6: Protótipo da Tela Inicial	29
Figura 7: Protótipo da Tela de Obras	29
Figura 8: Protótipo da Tela da Descrição	30
Figura 9: Protótipo da Tela das perguntas.....	30
Figura 10: Protótipo da Tela de Cadastro	31
Figura 11: Protótipo da Tela de Liderança	31
Figura 12: Protótipo da Tela de Avaliação do Museu	32
Figura 13: Protótipo da Tela de Agradecimento.....	33
Figura 14: Protótipo da Tela do Mapa	33
Figura 15: Protótipo da Tela de Relatório.....	34
Figura 16: Menu Principal	36
Figura 17: Tela do Mapa	36
Figura 18: Lista de Obras	37
Figura 19: Descrição da obra	38
Figura 20: Questionário	39
Figura 21: Cadastro com Teclado.....	40
Figura 22: Cadastro com Teclado Numérico.....	40
Figura 23: Avaliação Nota 1 a 5	41
Figura 24: Avaliação comentário opcional	41
Figura 25: Estatísticas - Respostas do Usuário	42
Figura 26: Estatísticas - Estatística Global.....	43
Figura 27: Tabela de Liderança	44
Figura 28: Tela de Agradecimento	44
Figura 29: Gerenciador de Usuários.....	45
Figura 30: Relatório	47
Figura 31: Relatório das Perguntas.....	47
Figura 32: Exemplo - Serialização de Objeto	48
Figura 33: Exemplo - Deserialização de Objeto.....	49
Figura 34: Arquivos do Sistema	137
Figura 35: Diagrama de Classe - Mapa e ListaObras.....	138
Figura 36: Diagrama de Classe - Sequência Questionario	139
Figura 37: Diagrama de Classe - Salvar Usuário	140
Figura 38: Diagrama de Classe - Sequência Final do Questionario/ Deserialização.....	141
Figura 39: Diagrama de Classe - Relatório.....	142
Figura 40: Obra Astronauta Neil Armstrong.....	143
Figura 41: Obra Astronauta Edwin E. Aldrin Jr.....	144
Figura 42: Obra Astronauta Michael Collins.....	144
Figura 43: Obra A tripulação da Apollo 11	145
Figura 44: Obra Saturn V Ventures	146
Figura 45: Obra Programa Saturno Apollo	147
Figura 46: Obra Módulo Lunar Apollo 11.....	148

Figura 47: Obra Módulo Lunar da Apollo 11 na superfície lunar	148
Figura 48: Obra Pegada no solo lunar	149
Figura 49: Obra Vista da lua	150

LISTA DE TABELAS

Tabela 1: Orçamento	16
Tabela 2: Matriz de Responsabilidade.....	17
Tabela 3: Cronograma Parte 1	18
Tabela 4: Cronograma Parte 2	18

SUMARIO

1.	INTRODUÇÃO	10
1.1	OBJETIVO GERAL	11
1.2	OBJETIVOS ESPECÍFICOS	11
2.	PESQUISA BIBLIOGRÁFICA.....	11
3.	CONTEXTUALIZAÇÃO DO PROJETO.....	12
4.	PLANEJAMENTO	14
4.1	TERMO DE ABERTURA DE PROJETO (TAP).....	14
4.2	ORÇAMENTO DO PROJETO	15
4.3	RESPONSABILIDADE DOS MEMBROS DA EQUIPE.....	16
4.4	CRONOGRAMA	17
4.5	DESENVOLVIMENTO	18
4.6	QUALIDADE DE SOFTWARE	19
4.7	CICLO DE VIDA	20
4.8	REQUISITOS FUNCIONAIS.....	21
4.9	REQUISITOS NÃO FUNCIONAIS.....	22
4.9.1	Método Ágil.....	22
4.9.2	User Interface e User Experience.....	23
4.9.3	Implementação do Dispositivo.....	23
4.9.4	Ferramentas para equipe	25
4.9.5	Ferramentas para Desenvolvimento	25
4.9.6	Banco de Dados.....	25
4.9.7	Redes De Computadores.....	26
4.10	PROTÓTIPO	28
5.	SOFTWARE.....	35
5.1	MANUAL DE INSTALAÇÃO.....	35
5.2	INTRODUÇÃO AO SISTEMA	35
5.3	SEQUÊNCIA DE TELAS.....	36
5.3.1	Janela Escondida	44
5.3.2	Relatório	46
5.3.3	Teclado	47
5.3.4	Imagens	48
5.3.5	Newtonsoft.Json.....	48
5.3.6	Tempo de Ausência	49

5.4 MANUAL DE DESINSTALAÇÃO	49
CONCLUSÃO	51
BIBLIOGRAFIA.....	52
APÊNDICE A - PERGUNTAS E RESPOSTAS.....	54
APÊNDICE B – FORM1 (MENU PRINCIPAL)	56
APÊNDICE C – MAPA.....	59
APÊNDICE D – LISTA OBRAS.....	60
APÊNDICE E – OBRASDESCRICOES.....	64
APÊNDICE F – QUESTIONÁRIO.....	67
APÊNDICE G – CADASTROUSUARIO	72
APÊNDICE H – AVALIACAOMUSEU	89
APÊNDICE I – ESTATISTICAS	107
APÊNDICE J – TABELADELIDERANCA	116
APÊNDICE K – AGRADECIMENTO	118
APÊNDICE L – RELATORIO	119
APÊNDICE M – RELATORIOPERGUNTAS.....	121
APÊNDICE N – GERENCIADORUSUARIOS	125
APÊNDICE O – CONTROLE	127
APÊNDICE P – VALIDACAO.....	131
APÊNDICE Q – ESTATICO	133
APÊNDICE R – JSONCONTROLE	135
APÊNDICE S – USUARIOS	136
APÊNDICE T – ARQUIVOS DO SISTEMA	137
APÊNDICE U – DIAGRAMAS DE CLASSE.....	138
ANEXO A - OBRAS E DESCRIÇÕES	143

1. INTRODUÇÃO

Atualmente a tecnologia é usada de diversas maneiras e em vários setores na sociedade, a evolução da tecnologia vem permitindo uma grande conexão causando mudança na forma como as pessoas vivem em seu dia a dia, trazendo comodidade e facilidade.

Quando se fala em museu, a mente se volta naturalmente para imagens de exposições de arte, relíquias e eventos históricos, existem diversos tipos de museus, no entanto, um museu multitemático abriga uma variedade de coleções e exposições relacionada a diferentes temas e períodos históricos. Ao contrário dos museus especializados, que estão concentrados em uma área específica como história natural ou arte contemporânea, os museus multitemáticos oferecem uma ampla gama de conhecimentos para os visitantes.

A chegada da tecnologia transformou significativamente a forma como se interage com espaços culturais, como o museu, cada vez que a tecnologia evolui, pode-se pensar em várias experiências que pode ser reimaginada, desde um aplicativo que funciona como um dicionário até uma experiência interativa de acampar em realidade aumentada, a tecnologia não só simplifica as interações como também mostra novas perspectivas para interpretação e comunicação.

O Instituto da Cultura e Ciência (ICC), que é uma organização sem fins lucrativos, dedicado a promover o conhecimento da cultura e da ciência em comunidades carentes no estado de São Paulo, tomou a iniciativa de criar um museu multitemático, que apresentará uma exposição sobre a ida do homem à lua.

Com o objetivo de proporcionar uma experiência mais enriquecedora e acessível aos visitantes, surge o projeto de um totem interativo para auxiliar os visitantes durante o passeio no museu.

Este projeto visa integrar a inovação tecnológica com preservação do conhecimento histórico e cultural relacionado à exploração espacial, sobre a jornada do homem à lua, ao unir elementos de interatividade, informação e imersão, o totem busca oferecer uma experiência dinâmica e educativa aos visitantes do museu.

O desenvolvimento e a implementação do software para o totem interativo, será detalhado desde a definição dos objetivos, recursos e funcionalidades, assim como os aspectos tecnológicos que moldarão a experiência do usuário.

1.1 OBJETIVO GERAL

Desenvolver um software para um totem interativo destinado aos visitantes de um museu multitemático, que proporcionará aos visitantes conhecimento e informação permitindo explorar e compreender a jornada do homem na lua.

1.2 OBJETIVOS ESPECÍFICOS

Esse projeto tem como objetivos específicos:

- Analisar os requisitos solicitados pelo cliente
- Planejar o sistema de totem
- Criar protótipo do totem
- Planejar o banco de dados
- Desenvolver o sistema do totem
- Criar Manual

2. PESQUISA BIBLIOGRÁFICA

Para o desenvolvimento do projeto foi realizado uma pesquisa bibliográfica, através de livros, artigos científicos e publicações acadêmicas, possibilitando uma melhor compreensão das práticas e teorias do desenvolvimento de software, sistemas interativos e tecnologias de interface usuário-máquina, permitindo compreender os conceitos e metodologias para o desenvolvimento do projeto.

De acordo com Gil (2002), a pesquisa bibliográfica é um tipo de pesquisa que se baseia na análise e interpretação de fontes secundárias, que proporciona uma base sólida de conhecimento teórico, permitindo uma compreensão aprofundada dos conceitos para o desenvolvimento do projeto.

3. CONTEXTUALIZAÇÃO DO PROJETO

O Instituto da Cultura e Ciência (ICC), é uma organização sem fins lucrativos, pretende criar um museu multitemático com exposição referente à primeira viagem do homem à lua, em comemoração aos 55 anos.

O instituto (ICC), pretende proporcionar experiências educativas para o público promovendo conhecimento gerando curiosidade e imaginação, para tornar a experiência mais interessante aos visitantes, o (ICC), entrou em contato com a equipe LIT - tecnologia, conhecida por seu compromisso com trabalhos e inovação tecnológica, para que através de uma parceria torne a experiência mais interativa para os visitantes no museu.

A equipe LIT - tecnologia, se comprometeu a desenvolver um software personalizado para usar em totens interativos para serem usados no museu incentivando os visitantes a responderem uma pesquisa de múltipla escolha sobre a exposição, através do resultado da pesquisa realizada, o totem fornece relatórios que irá permitir a instituição decidir por quanto tempo o tema da exposição será exposto, se será necessário fazer melhorias ou encerrar a exposição.

O instituto (ICC), conta com o apoio de stakeholders para a realização da exposição sobre a viagem do homem à lua, que incluem empresas locais, instituições educacionais, autoridades governamentais e organizações culturais, o envolvimento desses stakeholders são fundamentais para garantir o sucesso da exposição, fornecendo recursos para a realização do evento no museu, assim como o desenvolvimento do software para o totem.

De acordo com Pereira (2021), um museu é um local que preserva objetos históricos em exposição para manter a cultura, história de um local, da região ou do mundo oferecendo conhecimento, aprendizado e reflexão, através do decorrer do tempo, existem museus específicos e museus multitemáticos, que apresenta vários temas, coleções e exposições no mesmo local.

No dia 20 de julho de 1969 ocorreu um marco histórico para a conquista espacial, avanço científico e tecnológico: a viagem do homem à lua. O astronauta Neil Armstrong e seu colega de missão Edwin Aldrin passaram cerca de 22 horas na lua, dessas, somente 2h40 fora da nave. (PEREIRA, 2021).

O primeiro homem a pisar em solo lunar foi o astronauta norte-americano Neil Armstrong, em uma conquista obtida pelo Estados Unidos, que concorria na época com a antiga União Soviética em meio a Guerra Fria, pelo poder científico e tecnológico. (CIÊNCIA E CULTURA, p.10 2009).

A viagem a lua trouxe um grande número de avanços tecnológicos, tanto para áreas como a medicina, tanto para o dia a dia da humanidade, como por exemplos as frigideiras teflon. Os satélites espaciais hoje são de extrema importância para a humanidade, hoje são utilizados em serviços de internet, monitoramento de clima, desmatamento, entre outros. (FRIEDLAENDER, 2024).

Novas viagens para a lua são planejadas pelos cientistas, a fim de responder as perguntas que ainda estão sem respostas, como por exemplo, entender a origem do nosso planeta. No ano de 2024 a NASA planeja retornar a lua com o objetivo de iniciar um processo de colonização, para a longo prazo, se tornar um ponto de partida para viagens a marte. (FRIEDLAENDER, 2024).

A exposição no museu irá contar a história da viagem do homem à lua, o contexto político e social da época, mostrando o legado dessa conquista para a humanidade, o desenvolvimento de um software para ser usado nos totens durante a exposição, deixará a exposição dinâmica e interessante para os visitantes, além de contribuir com o (ICC), com informações importantes para as próximas exposições.

A ideia para o desenvolvimento de um software para um totem interativo no museu multitemático com exposição sobre o tema: “O homem na lua”, é proporcionar ao usuário uma experiência de interação e conhecimento para os visitantes, dentro da história da missão do Apollo 11.

Com uma interface intuitiva o totem possui informações sobre as obras, questionários, conteúdo multimídia e um mapa do museu, a fim de inserir a tecnologia no ambiente de maneira interativa e educativa e de fácil acesso aos visitantes.

A implantação de totens interativos oferece uma interação proporcionada pela tecnologia, deixando a experiência mais dinâmica e imersiva, além da sua funcionalidade de colher informações para avaliar a aceitação da exposição e identificar áreas para melhorias.

4. PLANEJAMENTO

De acordo com Sommerville (2011), para que um projeto ou até mesmo uma ação seja executada corretamente, é preciso fazer um planejamento antes de tudo, sendo muito importante para o sucesso de qualquer projeto, o planejamento torna o projeto não só eficiente, mas rápido para ser desenvolvido e entregue ao cliente, garantindo a correção e a otimização dos recursos com qualidade.

4.1 TERMO DE ABERTURA DE PROJETO (TAP)

De acordo com PMI (2017), o termo de abertura de projeto, é um documento formal que marca o início do projeto elaborado pela equipe responsável do projeto em conjunto com a parte interessada, ou seja, o solicitante do projeto. O TAP, descreve os recursos, os objetivos, os requisitos iniciais, o cronograma inicial, os riscos identificados e as informações necessárias, serve como um contrato interno que estabelece as bases e as expectativas das partes interessadas no desenvolvimento do projeto, sendo importante para manter as partes alinhadas e garantir a compreensão de seus objetivos.

Figura 1: Termo de Abertura de Projeto

TERMO DE ABERTURA DE PROJETO (TAP)**Título do Projeto**

Projeto Totem Interativo – Museu Multitemático: “O Homem na Lua”

Justificativa

A criação de um totem interativo para o museu visa proporcionar uma experiência imersiva e educativa aos visitantes. Com a integração de tecnologia, o museu poderá modernizar sua abordagem de apresentação, permitindo uma interação mais dinâmica com os artefatos e informações expostas. Isso atrairá um público mais diversificado e aumentará o engajamento dos visitantes com o acervo.

Objetivos

- Desenvolver um software intuitivo e atrativo para o totem interativo.
- Melhorar a experiência do visitante no museu, fornecendo informações adicionais e interativas sobre os artefatos.
- Facilitar a compreensão e contextualização do acervo por parte dos visitantes, incluindo crianças, jovens e adultos.
- Reforçar a identidade e a imagem do museu como um local moderno e inovador.

Benefícios

- Aumento do engajamento dos visitantes com o acervo do museu.
- Ampliação do público-alvo, incluindo crianças, jovens e adultos.
- Maior retenção de visitantes, que passarão mais tempo explorando o museu.
- Potencial para atrair mídia e cobertura adicional devido à inovação tecnológica.

Principais Marcos

- Concepção e design do software do totem interativo.
- Desenvolvimento do protótipo funcional do totem.
- Testes de usabilidade e ajustes no software.
- Implementação do totem no museu.
- Lançamento oficial do totem interativo para o público.

Premissas

- Disponibilidade de recursos financeiros para o desenvolvimento e implementação do projeto.
- Colaboração efetiva entre a equipe de desenvolvimento de software e a equipe do museu.
- Acesso a materiais e informações relevantes sobre o acervo do museu para integração no software.

Restrições

- Limitações de tempo para o desenvolvimento e implementação do projeto.
- Restrições de espaço físico no museu para a instalação do totem.
- Necessidade de compatibilidade com as tecnologias existentes no museu, como sistemas de segurança e infraestrutura de rede.

Fonte: Autoria Própria

4.2 ORÇAMENTO DO PROJETO

Para dar início ao projeto do totem interativo, foram realizadas algumas pesquisas de mercado e estudos de viabilidade para entender as necessidades do museu e as expectativas do público visitante, em seguida foi estabelecido um orçamento adequado para o desenvolvimento e implantação do totem interativo, como está na tabela a seguir, este orçamento inclui os custos relacionados ao desenvolvimento do software e dispositivos que serão usados no museu, mão de obra e outros custos para rede ou alguma manutenção.

Tabela 1: Orçamento

ORÇAMENTO		
Desenvolvimento de Software	R\$	81.600,00
Dispositivos	R\$	13.659,02
Mão de Obra	R\$	13.500,00
Outros Custos	R\$	8.000,00
TOTAL	R\$	116.759,02

Fonte: Autoria Própria

Desenvolvimento de Software: Neste projeto, foi colocado quatro membros da Equipe LIT para fazer o planejamento do software, toda documentação e o software para o totem. Durante esses três meses, cada membro terá seu pagamento de R\$6.800 para desenvolver e documentar.

Dispositivos: Para este museu, foi colocado dois totens da Index Soluções, cada um custando R\$6.829,51.

Mão de Obra: Para a instalação dos dois totens, foi contratado três membros da equipe técnica, o preço total para a equipe será de R\$13.500.

Outros Custos: Foi colocado R\$8.000 para questões de cabos de redes, modem, switch ou até mesmo alguma manutenção do dispositivo.

4.3 RESPONSABILIDADE DOS MEMBROS DA EQUIPE.

Para organizar o projeto, definindo a função de cada membro da equipe foi utilizada a matriz RACI para distribui as responsabilidades da equipe, conforme apresentado na figura 1.

A matriz RACI é uma ferramenta que auxilia na gestão de projetos ajudando a definir e comunicar os papéis e responsabilidades das atividades para cada membro da equipe evitando conflitos, ela apresenta quatro tipos de responsabilidades que podem ser atribuídas para cada tarefa, atividade ou decisão:

Responsável é aquele que tem a responsabilidade pela execução das atividades;

Aprovador tem a autoridade para tomar decisões e garantir que o trabalho seja feito corretamente.

Consultado é aquele que os membros recorrem para durante o processo, mas não tem autoridade para aprovar ou executar a atividade.

Informado é aquele membro que precisa receber as informações sobre o projeto para se manter atualizado, mas não está diretamente envolvido na execução.

A matriz RACI organiza através de uma tabela onde as atividades são listadas em linhas e os papéis em colunas, sendo preenchidas com as iniciais correspondentes de cada responsabilidade.

Tabela 2: Matriz de Responsabilidade

ATIVIDADES	LEANDRO MORAES	THAIS REIS	GUSTAVO HENRIQUE	ISAC FERREIRA
Introdução	R	I	I	I
Objetivos	R	I	I	I
Contextualização do Caso	A	R	I	I
Desenvolvimento	A	R	I	I
Requisitos Funcionais	R	I	I	I
Requisitos não Funcionais	R	I	I	I
Protótipo	A		I	R
UI/UX	A		I	R
Banco de Dados	A		R	I
Redes	A	R	I	I
Frontend	R		I	C
Backend	R	I	I	I
Diagramas	R		I	I
Testes	R	I	I	I

Fonte: Autoria Própria

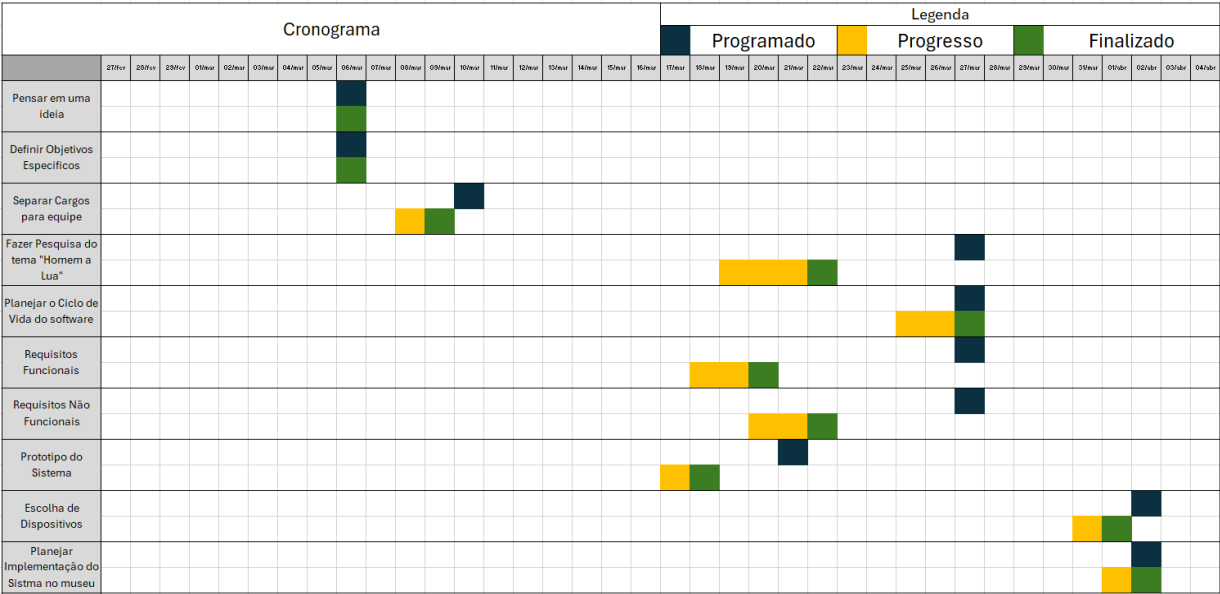
4.4 CRONOGRAMA

Um cronograma é uma representação visual, que apresenta as tarefas que foram definidas, e os prazos estabelecidos para entrega, fornecendo uma estrutura organizada para o planejamento e execução do projeto.

Após definir a função de cada membro na equipe se faz necessário montar um cronograma para alcançar as metas estabelecidas no projeto, garantindo seu desenvolvimento nos prazos estabelecidos, possibilitando manter a organização e facilitando a comunicação da equipe.

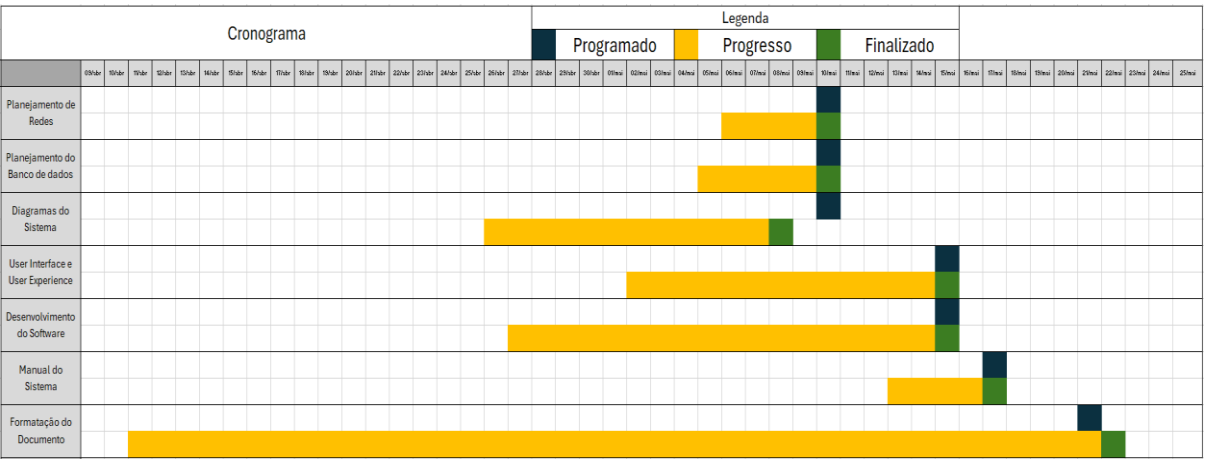
O cronograma da equipe se encontra nas tabelas 3 e 4:

Tabela 3: Cronograma Parte 1



Fonte: Autoria Própria

Tabela 4: Cronograma Parte 2



Fonte: Autoria Própria

4.5 DESENVOLVIMENTO

De acordo com Pressman (2020), antes de iniciar o desenvolvimento de um software é preciso analisar os requisitos do sistema, identificando as necessidades do cliente, as expectativas e as limitações técnicas do ambiente, para poder selecionar as tecnologias apropriadas para o desenvolvimento do sistema, após ter a base de hardware design de UI/UX estabelecidos, é possível começar o desenvolvimento de software, com criação de aplicativos, integração de sistema de gerenciamento de

conteúdo informações e desenvolvimento de funcionalidades específicas. Após o desenvolvimento inicial, o sistema passa por testes de qualidade para garantir que todas as funcionalidades operem de acordo com o esperado, sempre com ajustes e melhorias para garantir uma experiência otimizada, com manutenção contínua garantindo que o sistema permaneça operacional durante o tempo necessário.

4.6 QUALIDADE DE SOFTWARE

De acordo com Pressman (2020), ISO 9126 é uma norma internacional que estabelece diretrizes e critérios para avaliar a qualidade do software, composta por duas partes, a norma descreve um modelo de qualidade do software na primeira parte, definindo as características de qualidade e atributos associados, que são a funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade, essas características são importantes para avaliar a qualidade do software.

Ainda de acordo com o autor citado acima, a segunda parte da ISO 9126, fornece diretrizes para a avaliação do modelo e define métricas específicas para medir a qualidade do software em cada uma das características citadas na primeira parte, essas métricas ajudam a avaliar o grau de qualidade do software oferecendo orientações práticas para aplicação.

De acordo com a norma ISO 9126, as características estão descritas abaixo e na figura a seguir:

Funcionalidade: corresponde as capacidades fornecidas pelo software sobre os requisitos especificados, incluindo sua adequação funcional, precisão, e conformidade com os padrões.

Confiabilidade: é a característica que diz respeito à capacidade do software de manter seu nível de desempenho quando usado em condições estabelecidas durante um período determinado, incluindo capacidade de recuperação e estabilidade.

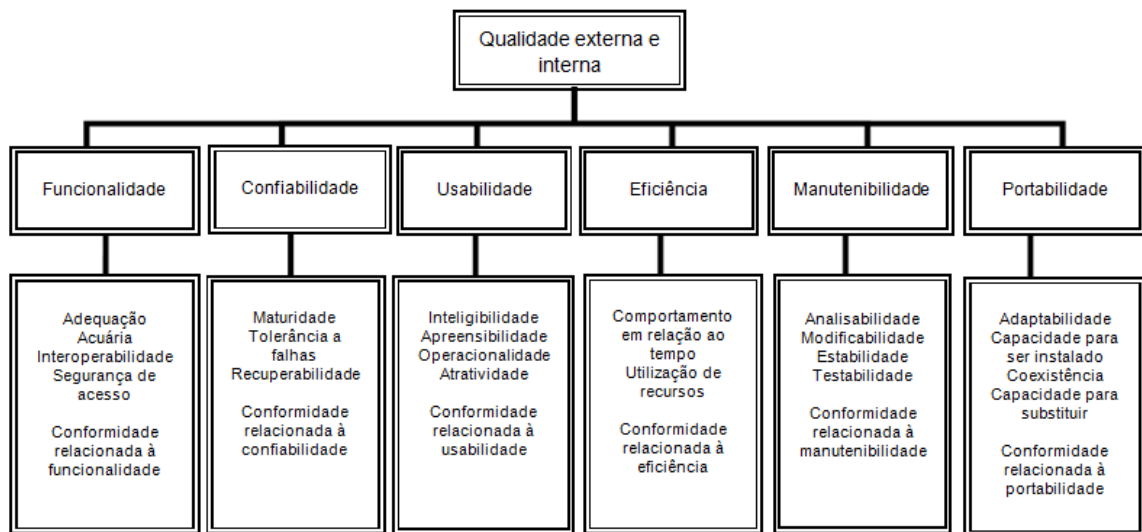
Usabilidade: refere-se à facilidade de uso do software pelo usuário, podendo dizer se considera fácil de interagir, compreender e operar.

Eficiência: avalia o desempenho do software com relação aos recursos utilizados como o consumo de memória, velocidade de processamento, e os recursos usados em condições estabelecidas.

Manutenibilidade: diz respeito a facilidade com que o software pode ser modificado, feito diagnóstico do que precisa ser corrigido e adaptado conforme os requisitos necessários.

Portabilidade: avalia a capacidade do software de poder ser transferido de um ambiente para outro e de se adaptar a diferentes plataformas e sistemas operacionais.

Figura 2: Princípios da qualidade de software da norma ISO 9126



Fonte: <https://www.researchgate.net/publication/332822299/figure/fig5/AS:754285969162240@1556847197082/Figura-5-Modelo-de-qualidade-externa-e-interna-ISO-IEC-9126.ppm>. Acessado em: 30 de abril de 2024

A ISO 9126 se destaca como uma ferramenta essencial para garantir a qualidade do software em diferentes contextos. Através de um modelo estruturado e métricas específicas, a norma facilita a comunicação entre stakeholders, promove o desenvolvimento de software de alto nível e contribui para o sucesso de projetos de software.

4.7 CICLO DE VIDA

De acordo com Kochanski (2013), o ciclo de vida é uma etapa importante de engenharia de software, essa é a etapa onde são decididos os passos de um sistema desde a sua criação até a finalização, ele servirá como guia para planejamento, desenvolvimento e manutenção do sistema. A qualidade do software durante o ciclo

de vida é importante para garantir que o produto final atenda às necessidades e expectativas dos usuários, desde a concepção, manutenção até a evolução.

Conforme explica o autor citado acima, é preciso garantir a qualidade em cada fase do ciclo de vida do software desde a concepção com definições claras dos requisitos do software, acompanhando o projeto e o desenvolvimento, realizando testes para identificar e corrigir defeitos precocemente, garantindo a implantação adequada do software e realizando manutenção para atender as necessidades dos usuários.

O ciclo de vida de um software inicia no momento em que são registradas as primeiras especificações de funcionamento, desenvolvimento, implementação, teste e lançamento, até o momento em que o software deixa de ser utilizado pelos seus usuários. (KOCHANISKI, p. 52 2013)

O instituto (ICC) está fazendo uma exposição referente a primeira viagem do homem à lua, que concluirá 55 anos no dia 20 de julho, então para que o sistema acompanhe o evento, e que não fique fora de data, o sistema ficará ativo por três meses, acabando no dia 25 de agosto, neste período o sistema terá as atualizações necessárias como melhoria de desempenho e correções de erros no sistema além da manutenção necessária. Através dos feedbacks será possível avaliar se a exposição teve uma boa aceitação, em caso positivo o sistema ficará ativo por mais dois meses.

4.8 REQUISITOS FUNCIONAIS

Requisitos funcionais. São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer. (SOMMERVILLE, 2011).

As funções planejadas para o totem são:

- Visualizar mapa do museu
- Selecionar obras
- Fazer um questionário
- Avaliar o museu

O totem tem um mapa que mostra as localizações de cada obra ao mesmo tempo, em que exibe a posição do usuário, facilitando a busca pela obra desejada. Caso o usuário tenha interesse em alguma das obras do museu, ele pode selecioná-la no totem, em seguida, a descrição aparecerá na tela, permitindo que o usuário saiba mais sobre a obra que está sendo apresentada.

Para os usuários que gostariam de colocar seu conhecimento em prática, o totem disponibiliza um questionário com mais de cinco perguntas sobre o evento. Ao completar as perguntas, o totem solicitará o nome e a idade do usuário, em seguida, mostrará a tabela de pontuações, que apresentará a posição do usuário, seu nome e sua pontuação.

Após terminar o questionário, os usuários podem compartilhar suas impressões que tiveram no evento, na tela de avaliação, podem atribuir uma pontuação de 1 a 5 luas (em homenagem ao tema do museu), e optar por deixar um comentário opcional que sistema irá guardar para os funcionários avaliarem.

4.9 REQUISITOS NÃO FUNCIONAIS

Requisitos não funcionais. São restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo. (SOMMERVILLE, 2011).

4.9.1 Método Ágil

[...]Métodos ágeis, universalmente, baseiam-se em uma abordagem incremental para a especificação, o desenvolvimento e a entrega do software. Eles são mais adequados ao desenvolvimento de aplicativos nos quais os requisitos de sistema mudam rapidamente durante o processo de desenvolvimento. Destinam-se a entregar o software rapidamente aos clientes, em funcionamento, e estes podem, em seguida, propor alterações e novos requisitos a serem incluídos nas iterações posteriores do sistema. Têm como objetivo reduzir a burocracia do processo, evitando

qualquer trabalho de valor duvidoso de longo prazo e qualquer documentação que provavelmente nunca será usada. (SOMMERVILLE, 2011)

Para o início do projeto, optamos por adotar uma abordagem ágil de desenvolvimento de software. Esta escolha foi feita com o intuito de garantir uma entrega de um produto de qualidade dentro do prazo estabelecido. A metodologia permitiu a divisão do trabalho em tarefas curtas, conhecido como “Sprint”. Essa entrega contínua possibilitou uma avaliação constante por parte do cliente, permitindo feedbacks e ajustes rápidos conforme necessário.

4.9.2 User Interface e User Experience

Segundo Fabricio Teixeira (2014), experiência de usuário (UX/ User Experience) tem como objetivo principal garantir que os produtos sejam fáceis de usar, intuitivos e proporcionem uma experiência positiva para quem os utiliza. Isso envolve entender as necessidades e os comportamentos dos usuários, criar interfaces que atendam a essas necessidades de forma eficaz e garantir que a interação com o produto seja suave e agradável.

Ainda de acordo com Fabricio Teixeira (2014), interface de usuário (UI/ User Interface) tem como objetivo o processo de projetar a aparência e a interação de um produto digital, como um aplicativo ou um site. UI Design se concentra especificamente na interface visual e interativa que o usuário utiliza para interagir com o produto. Isso inclui o design de elementos como botões, menus, ícones, tipografia, cores e layouts, com o objetivo de criar uma interface intuitiva, atraente e eficaz que facilite a interação do usuário com o produto. Em resumo, o UI Design visa criar uma experiência visualmente agradável e funcional para os usuários.

4.9.3 Implementação do Dispositivo

Para o funcionamento do sistema de totem no museu, é necessário um dispositivo que irá hospedar, essa escolha é importante para que a equipe responsável, esteja preparado para instalar o dispositivo, configurar o software e se necessário fazer a manutenção, para corrigir as possíveis falhas. Para evitar filas e proporcionar uma experiência mais tranquila e sem pressa para os visitantes, foi decidido uma instalação de dois dispositivos no museu. Esses dois totens permitirão

que mais pessoas usem o sistema simultaneamente, contribuindo para uma experiência mais fluida e satisfatória durante a visita ao museu.

O produto escolhido foi o Totem Digital Touch Screen - IDX Smart, código: NCJVH8DR4 da marca Index Soluções, o totem possui:

- Tela de 18,5 (16:9) polegadas com touchscreen
- 4 Gigabyte de memória Ram (Random-Access Memory)
- Processador Intel core I3
- SSD (solid-state drive) 120 Gigabyte

As especificações técnicas estão de acordo com o planejamento, a tela de 18 polegadas com touchscreen para que o usuário consiga interagir com os botões do sistema, e para o sistema fluir sem nenhum travamento, o totem terá SSD, também fazendo com que o software seja mais rápido na hora do processamento.

Após a seleção do dispositivo, é preciso ter algumas etapas planejadas para realizar a instalação do totem nos museus, começando pela logística de transporte e chegando à montagem física do dispositivo no local escolhido dentro do museu, durante essa fase, serão consideradas as necessidades específicas de cada museu em termos de posicionamento e integração com a infraestrutura existente. Para a instalação física de dois totens, deve demorar aproximadamente 1 hora cada dispositivo, dependendo da complexidade do ambiente e dos requisitos de instalação.

Após a instalação física, a equipe técnica focará na instalação e configuração do software no dispositivo instalado, o processo vai passar desde a transferência dos arquivos de software até a configuração de utilização, em seguida terá testes das funcionalidades para garantir que tudo esteja conforme o esperado. Para a instalação e configuração completa do software, deve demorar aproximadamente 30 minutos, considerando o procedimento de testes e verificações.

Ao seguir essas etapas cuidadosamente, estaremos prontos para implementar os totens no museu, e prontos para criar uma experiência para os visitantes e fortalecendo tanto a cultura quanto a educação.

4.9.4 Ferramentas para equipe

Para uma organização na equipe, foi usado o sistema Trello que permitiu uma organização de objetivos, usando cartões para separar atividades e checklist para separar as tarefas.

Para comunicação e coordenação de pedidos, foi usado o Whatsapp. Ele proporcionou uma forma rápida e direta de troca de informações entre os membros da equipe, foi garantido uma sincronia e integração das atividades da equipe.

4.9.5 Ferramentas para Desenvolvimento

O Figma foi escolhido como uma ferramenta para desenvolver as telas do totem do museu, pois é um aplicativo de fácil manuseio, com diversas ferramentas disponíveis para ajudar no desenvolvimento, tendo a possibilidade de agregar mais pessoas dentro do projeto e todos poderem mexer ao mesmo tempo no aplicativo. Figma disponibiliza uma tela ampla, ferramentas de criações, modelos de telas já existente e um criador de sequência.

O Floorplanner é um site para desenvolver plantas de projetos, e foi escolhido para fazer o mapa do museu, pois tem um fácil manuseio, e diversas ferramentas, sendo possível criar a espessura da parede, tamanho dos cômodos, altura do local, colocar móveis no local, escolher cores das paredes. sendo a ferramenta ideal para desenvolver o mapa.

Para o desenvolvimento do sistema, foi usado Visual Studio 2022 Community usando a plataforma .NET versão 8, onde tem o modelo de projeto Windows Forms, nele é possível criar telas com mais facilidade. A escolha de usar o Visual Studio também vem da linguagem C# que é uma linguagem ampla e com várias bibliotecas de código aberto.

4.9.6 Banco de Dados

Neste projeto, optamos por utilizar o formato JSON (JavaScript Object Notation) para armazenar e gerenciar os dados. O JSON foi escolhido devido à sua facilidade de busca e manipulação de informações, bem como sua capacidade de armazenar dados de forma estruturada e legível.

O banco de dados em JSON é organizado em diversas coleções de dados, cada uma representando uma entidade ou conjunto específico de informações. O JSON foi usado para armazenar:

1. As informações de cadastro do usuário, que inclui Nome e Idade
2. Quantidade de acertos do usuário no questionário
3. Comentário opcional do usuário

Após o primeiro usuário concluir o questionário no sistema, as informações serão enviadas para um gerenciador de usuários e um arquivo JSON será criado no disco local C. Enquanto o sistema estiver ativo, os dados serão armazenados nesse JSON. No entanto, para economizar espaço de armazenamento, o sistema deleta o arquivo JSON ao ser fechado, garantindo que os dados não sejam persistidos no disco.

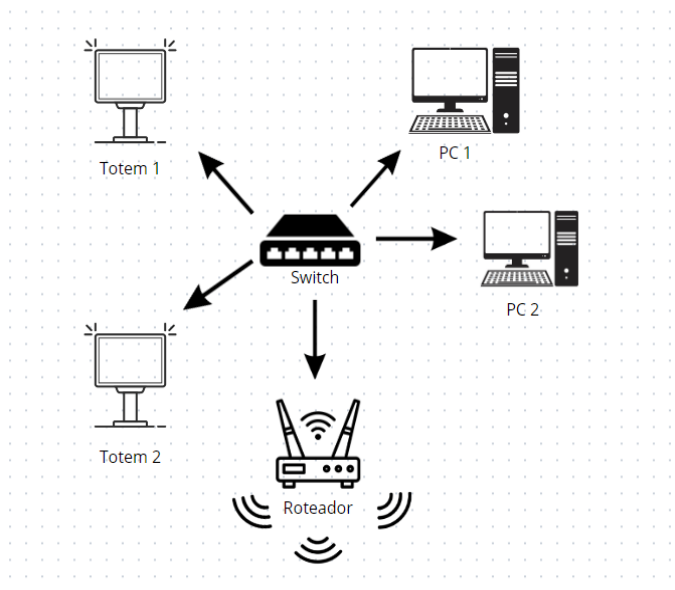
Para fazer esse processo de armazenamento de dados JSON em C#, utilizamos a biblioteca Newtonsoft.Json, uma biblioteca responsável por serializar e deserializar os dados JSON, simplificando o processo de leitura, escrita e manipulação dos dados do banco de dados

4.9.7 Redes De Computadores

O surgimento das redes de computadores revolucionou a organização dos sistemas computacionais. Antes, um único computador executava todas as tarefas de processamento e armazenamento de dados, devido ao alto custo das máquinas. Com avanços tecnológicos, os computadores se tornaram mais potentes, menores e acessíveis, permitindo sua disseminação. Isso levou a uma nova abordagem, onde múltiplos computadores são utilizados em sistemas distribuídos, em vez de depender de um único computador centralizado. (MACEDO et al, 2018).

No museu "O Homem na Lua", a infraestrutura de rede compreende: um switch que recebe internet fibra óptica da provedora, cabos de rede para internet conectados a dois totens interativos, dois computadores na recepção, uma impressora na recepção e um roteador Wi-Fi que distribui sinal sem fio por todo o museu. Arquitetado em infraestrutura lógica de estrela.

Figura 3: Infraestrutura de redes



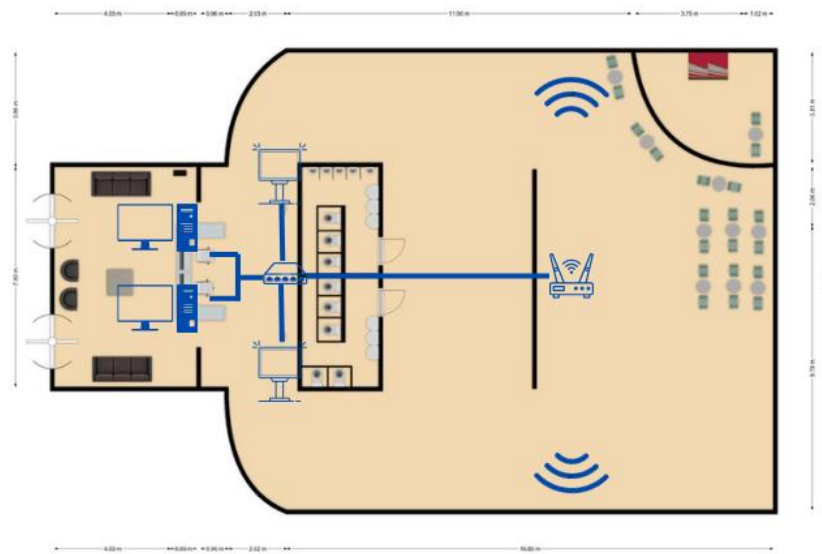
Fonte: Autoria própria

Na entrada do museu, foram instalados os seguintes equipamentos para garantir uma rede rápida e estável, proporcionando uma excelente experiência aos visitantes:

- **Cabo de rede Cat6 15m (R\$ 305,00):** Conecta os dispositivos, garantindo alta velocidade e estabilidade na transmissão de dados.
- **Switch de 8 portas TP-Link (R\$ 248,54):** Interliga os totens interativos e a recepção, permitindo uma distribuição eficiente da rede.
- **Roteador Wi-Fi 6 Gigabit Dual Band AX5400 (R\$ 699,00):** Posicionado no centro do museu, com seis antenas para irradiar a conexão Wi-Fi, proporcionando acesso rápido e confiável à internet para os visitantes.

Essa rede robusta permite que os visitantes explorem as obras de arte, interajam com os totens informativos e compartilhem suas experiências nas redes sociais.

Figura 4 - Mapa do museu com infraestrutura de redes



Fonte: Autoria própria

Figura 5 - Legenda de redes



Fonte: Autoria própria

4.10 PROTÓTIPO

A tela inicial do sistema é o Menu, contém as opções “Mapa”, “Obras” e “Questionário”, como está descrito na figura seguinte:

Figura 6: Protótipo da Tela Inicial



Fonte: Autoria Própria

Ao selecionar a opção “Obras”, irá aparecer a lista das obras do anexo A que estão sendo expostas no museu, como aparece na Figura 7:

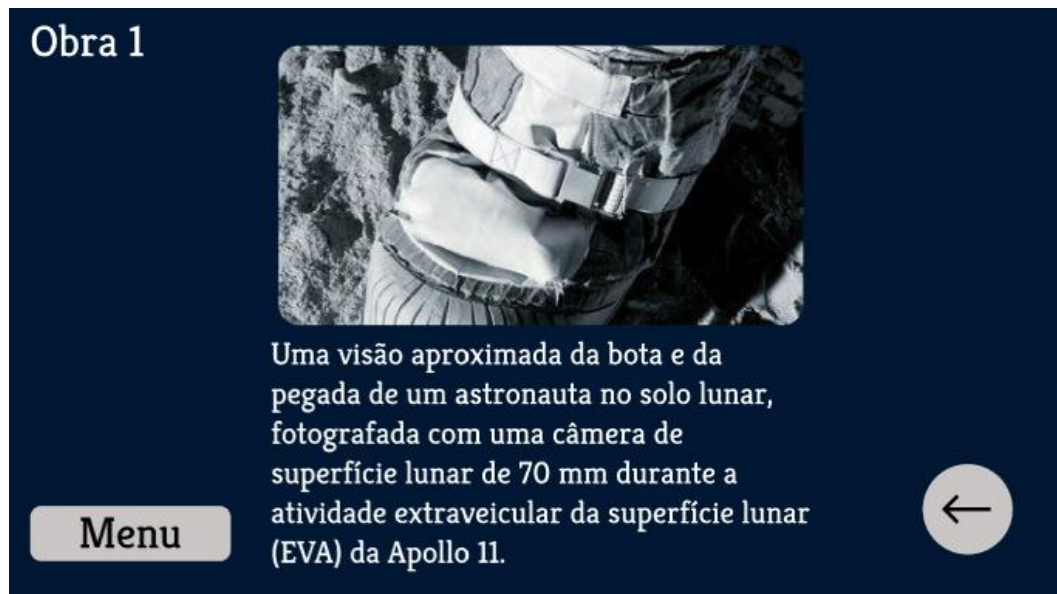
Figura 7: Protótipo da Tela de Obras



Fonte: Autoria Própria

Quando selecionado alguma obra, aparecerá uma tela com a imagem da obra e embaixo a descrição que está no anexo A junto das obras, como podemos ver na Figura 8:

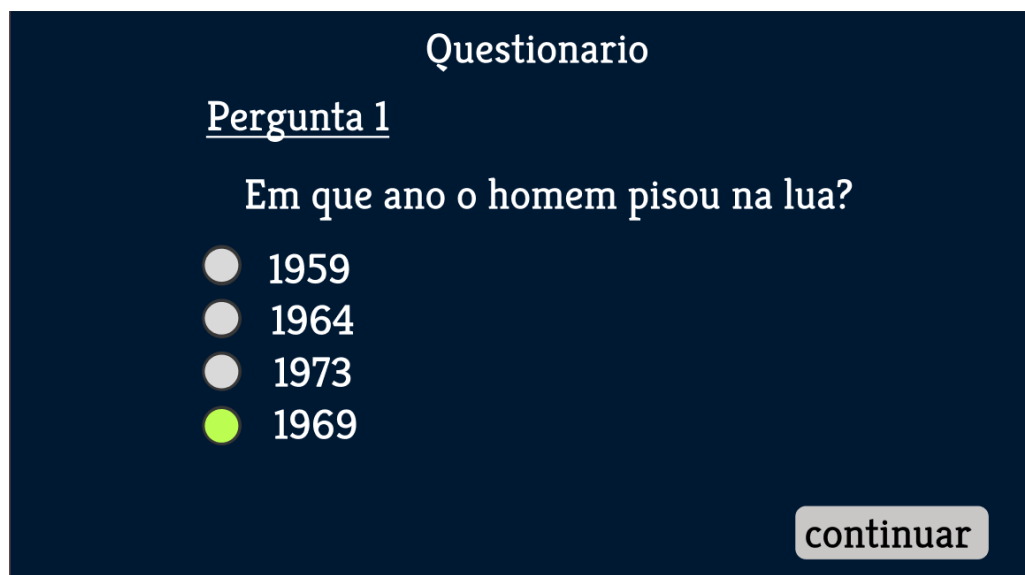
Figura 8: Protótipo da Tela da Descrição



Fonte: Autoria Própria

Após o cliente ler a obra que ele deseja, ele poderá colocar seu aprendizado em ação, ao voltar para o menu e clicar em "Questionário" ele terá cinco perguntas que estão no apêndice A para responder, todas de múltipla escolha como está na seguinte imagem:

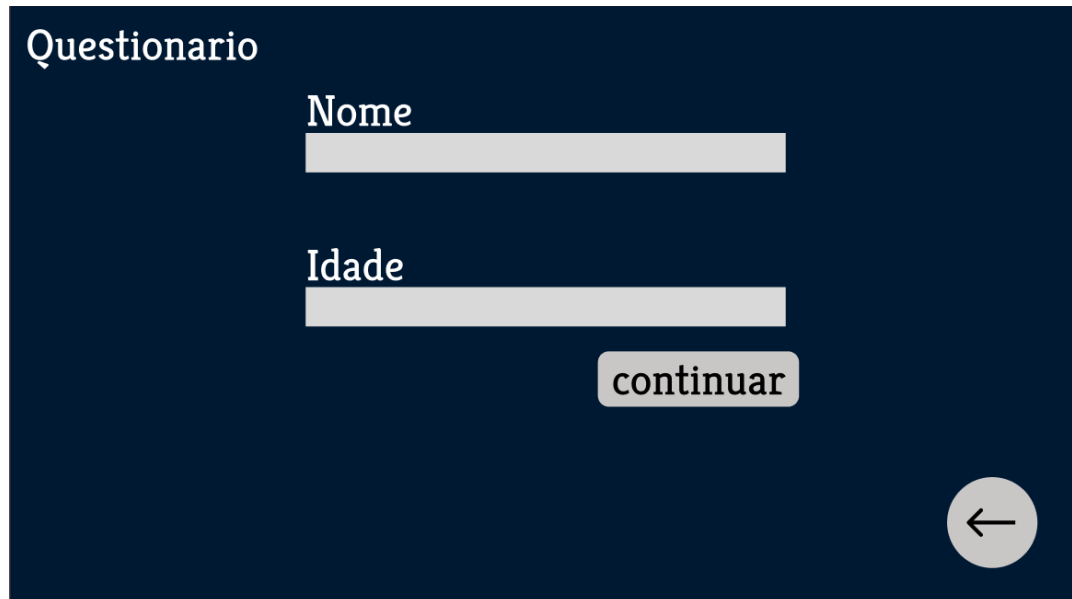
Figura 9: Protótipo da Tela das perguntas



Fonte: Autoria Própria

Quando o questionário estiver finalizado, o usuário poderá colocar seu nome e sua idade como está representado na Figura 10:

Figura 10: Protótipo da Tela de Cadastro



Questionario

Nome

Idade

continuar

←

Fonte: Autoria Própria

Após se cadastrar, seu nome aparecerá na tabela de liderança, onde vai aparecer também a colocação do visitante e sua pontuação como está na figura seguinte:

Figura 11: Protótipo da Tela de Liderança



TABELA DE LIDERANÇA

Colocação	Nome	Questões Acertadas
1º	Gabriel	5/5
2º	Maria Fernanda	5/5
3º	João	4/5
4º	Gustavo	3/5
5º	Jasmine	3/5

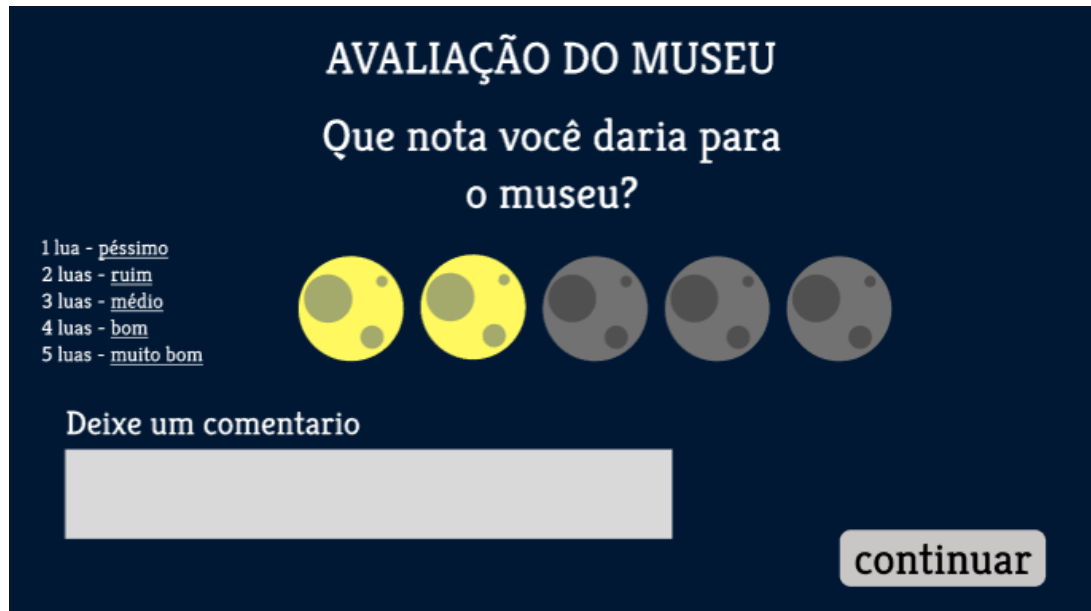
continuar

Fonte: Autoria Própria

Ao clicar em continuar, o sistema irá pedir a avaliação do usuário sobre o museu, a avaliação será uma nota de 1 a 5, usando luas para representar a nota, e

abaixo terá uma caixa de comentário, onde o usuário poderá escrever sua impressão que teve com o museu ou até mesmo com o totem, segue abaixo a representação da tela:

Figura 12: Protótipo da Tela de Avaliação do Museu



AVALIAÇÃO DO MUSEU

Que nota você daria para o museu?

1 lua - péssimo
2 luas - ruim
3 luas - médio
4 luas - bom
5 luas - muito bom

Deixe um comentario

continuar

The image shows a dark blue screen with white text. At the top, it says 'AVALIAÇÃO DO MUSEU' and 'Que nota você daria para o museu?'. Below this is a row of five circles representing moon phases. The first two are yellow, and the last three are grey. To the left of these circles is a list of ratings from 1 to 5, each with a description and a link. Below the circles is a text input field and a 'continuar' button.

Fonte: Autoria Própria

Após isso, quando o usuário clicar em continuar, vai aparecer a tela de agradecimento, como está representado na Figura 13, e depois voltara para o menu da Figura 6:

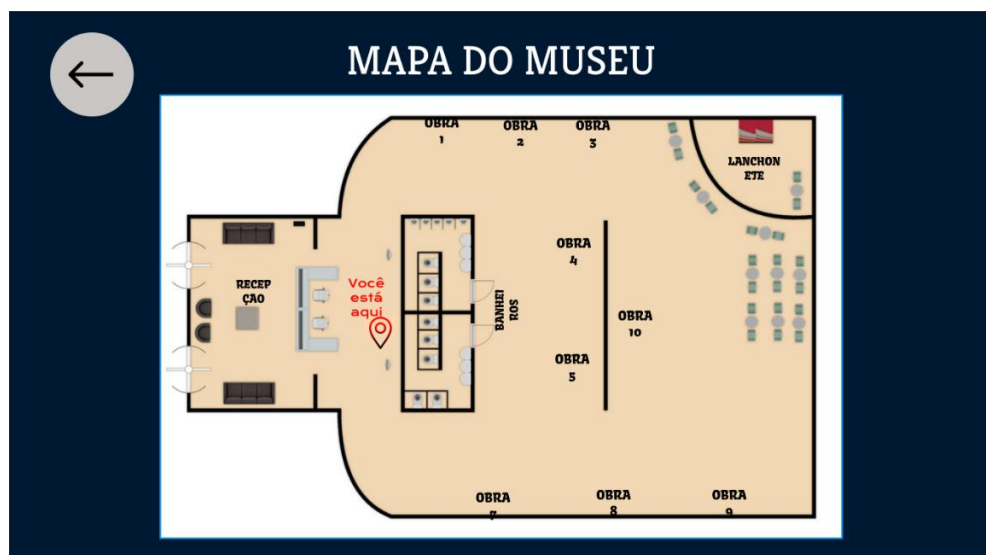
Figura 13: Protótipo da Tela de Agradecimento



Fonte: Autoria Própria

De volta para o menu, o usuário poderá clicar na opção “Mapa”, para se localizar e para localizar também as obras que ele deseja ver, contendo uma planta do local como está sendo representado na figura a seguir:

Figura 14: Protótipo da Tela do Mapa



Fonte: Autoria Própria

No final do expediente, os funcionários podem entrar na tela do relatório, na tela vai mostrar uma tabela com as informações que os visitantes preencheram na

Figura 10, e mais algumas informações, como as questões acertadas, nota do museu e o comentário do museu, que o visitante escreveu na figura 12, essas informações podem ajudar a tomar algumas tomadas de decisões futuras, segue abaixo o protótipo da tela de relatório:

Figura 15: Protótipo da Tela de Relatório

Menu		RELATÓRIO		
Nome	Idade	Questões Acertadas	Nota do Museu	Comentarios
Gabriel	21	5/5	5/5	"O museu é muito legal, obras muito lindas, aprendi muito hoje"
Maria Fernanda	18	5/5	5/5	"Achei muito bom, com certeza voltarei"
Luis	18	2/5	4/5	"Achei muito bom, porem podia ter mais obras"
Antonio	22	1/5	4/5	"Muito legal"

Fonte: Autoria Própria

5. SOFTWARE

O sistema desenvolvido tem como objetivo passar informação sobre a primeira viagem do homem à lua e incentivar os visitantes a fazer um questionário, junto disso o sistema gera relatório sobre as avaliações dos usuários com o museu e as perguntas.

5.1 MANUAL DE INSTALAÇÃO

A instalação do sistema de totem é um processo simples, sem a necessidade de programas adicionais em execução. O software é enviado em um arquivo compactado, que contém uma pasta denominada “SistemaTotem” com outros arquivos, conforme ilustrado na imagem do apêndice T.

A pasta deve ser movida para a Área de Trabalho e, em seguida, o sistema pode ser aberto através do arquivo executável presente na pasta.

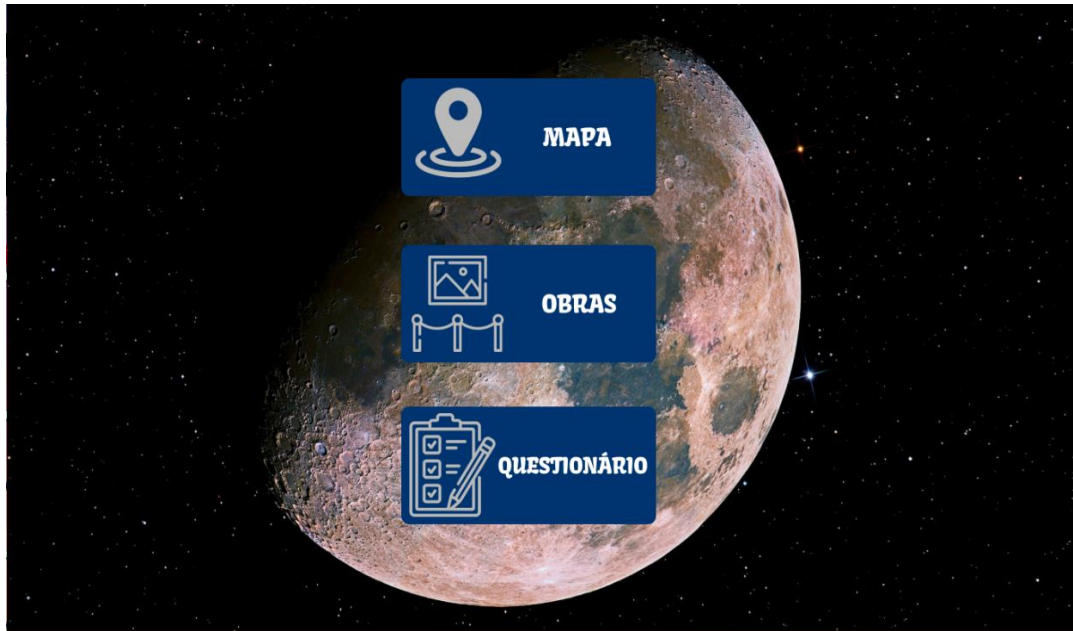
5.2 INTRODUÇÃO AO SISTEMA

As janelas do software são configuradas com uma resolução de 1366 pixels de largura e 768 pixels de altura, para que o sistema fique na proporção correta na tela do dispositivo do totem. As janelas não possuem bordas, impedindo que os usuários arrastem as janelas e dá uma sensação de que não foi adicionado uma janela a mais. A maioria das telas seguiu fielmente o que foi planejado no protótipo, mas algumas mudanças e adições foram necessárias no produto final para aprimorar a funcionalidade e a usabilidade.

Ao iniciar o sistema, o *Menu Principal (Form1)* é aberto junto com uma janela chamada *GerenciadorUsuarios*, que não é visível para os usuários. As opções do *Menu Principal* não foram modificadas no produto final, as opções continuam como:

- Mapa
- Obras
- Questionário

Figura 16: Menu Principal

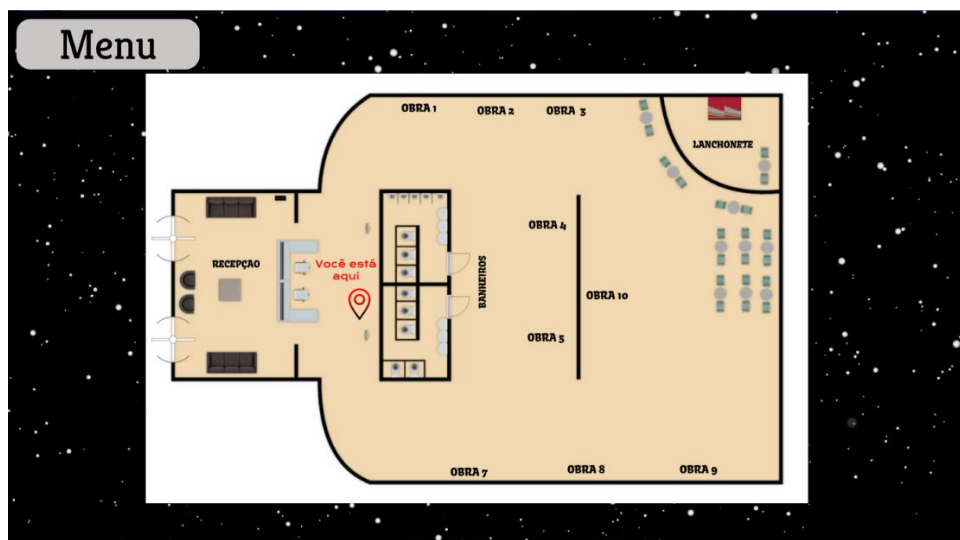


Fonte: Autoria Própria

5.3 SEQUÊNCIA DE TELAS

Como mencionado anteriormente, o usuário terá três opções no *Menu Principal*. A opção *Mapa* serve para o usuário se localizar e localizar as obras. Selecionando, uma nova janela é aberta com a visão do museu visto de cima, como está na figura a seguir:

Figura 17: Tela do Mapa



Fonte: Autoria Própria

A opção “*Obras*” abre uma janela nova que mostra uma lista de todas as obras que está sendo exposta no museu como está sendo representado na figura 18. O usuário tem a opção de navegar pelas obras usando as setas para avançar ou retroceder páginas. O sistema troca as imagens das obras por outras armazenadas conforme a navegação.

A tela de *Obras* contém duas *PictureBox* e dois *Labels* na tela. Quando a janela é iniciada, o método construtor chama a função *MudarObras* com o valor 1 na variável *paginaID*. A função atualiza as imagens das *PictureBox* e o texto dos *Labels* conforme o valor da *paginaID*. As setas seguem uma lógica semelhante: caso o usuário aperte na seta de avançar, o clique da seta adiciona mais 1 na variável *paginaID* e chama a função *MudarObras*. A mesma coisa se repete na seta de retroceder, mas ela subtrai o *paginaID*. Para evitar que as setas levem a uma página inexistente, o sistema desativa a seta ao chegar na página limite e se necessário o sistema redefine a *paginaID* para 1 e chama a função *MudarObras*.

Figura 18: Lista de Obras



Fonte: Autoria Própria

Ao clicar na obra desejada, o sistema abre uma janela nova chamada *ObrasDescricoes*, onde é exibido a imagem e a descrição da obra que o usuário selecionou, conforme é mostrado na figura 19

A lógica por trás disso é que, quando a função *MudarObras* é chamada é chamada, não apenas a imagem e o título mudam, mas também o Id da obra. Após

clicar em uma *pictureBox*, uma nova janela é aberta, passando o Id da obra para o método construtor. Quando o método construtor de *ObrasDescricoes* recebe o Id da obra, a função *AtualizarTela* é chamada. Esta função usa a variável em um *switch* para exibir a imagem, descrição e título da obra. Caso o Id da obra não seja encontrado, a janela *ObrasDescricoes* é fechada, retornando para a lista de obras.

Figura 19: Descrição da obra



Fonte: Autoria Própria

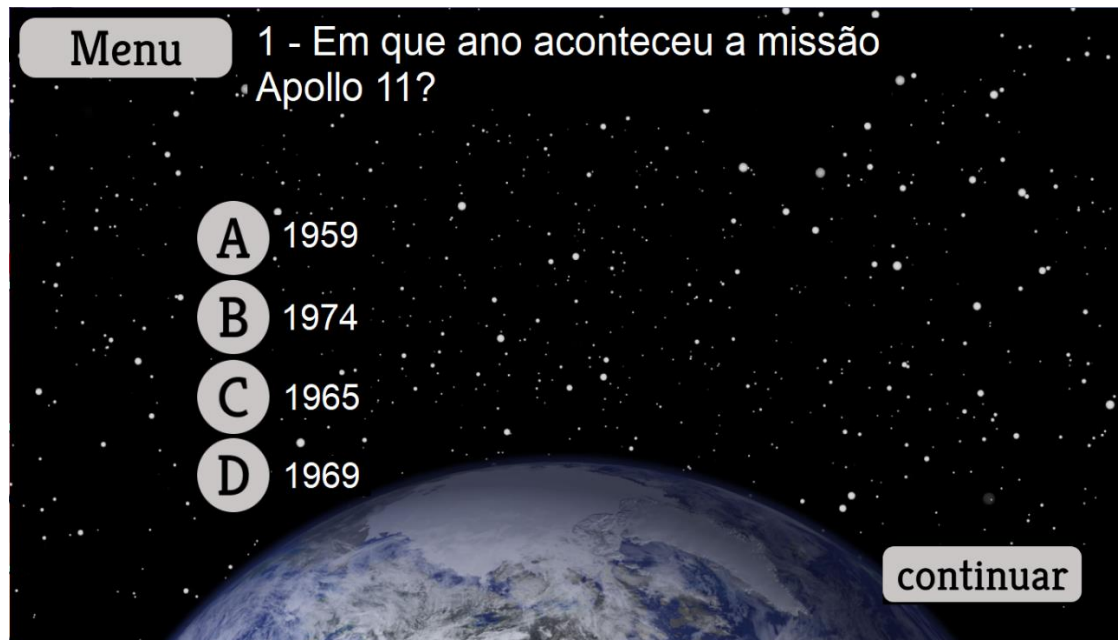
Na maioria das janelas, o botão *Menu* estará presente no mesmo local, permitindo que o usuário volte rapidamente ao *Menu Principal*. Após ler as descrições das obras, o usuário pode testar seus conhecimentos na opção *Questionario*.

Clicando na opção *Questionario* no *Menu Principal*, uma janela chamada *Questionario* é aberta, onde o usuário deve responder as cinco questões e quatro alternativas cada. O botão "Continuar" avança para a próxima pergunta, desde que uma alternativa tenha sido selecionada, conforme mostrado na figura 20.

A janela *Questionario* começa com o método construtor que define valores vazios para algumas variáveis estáticas e chama a função *ProximaPergunta* que começa recebendo o valor 1. A lógica da função *ProximaPergunta*, é semelhante à da função *MudarObras* na lista de obras. Ela recebe a variável *idPergunta* e atualiza a *Label* da pergunta e das alternativas. O botão *Continuar* segue a mesma lógica das setas na janela da lista de obras: ao ser pressionado cria um objeto *Controle* que valida a resposta selecionada e, se correta, incrementa a variável estática

QuestionarioNotaUsuario. Após a validação, o botão *idPergunta* e chama *ProximaPergunta* para continuar o questionário.

Figura 20: Questionário



Fonte: Autoria Própria

Após a quinta pergunta, o sistema chamar a tela janela *CadastroUsuario*, onde o usuário pode cadastrar seu nome e idade. Por ser um sistema para um totem touchscreen sem teclado físico, é necessário de um teclado virtual para que o usuário insira suas informações, como é representado nas figuras 21 e 22.

Nesta janela possui dois teclados, um teclado normal no formato *QWERTY* e um teclado numérico. Caso o usuário aperte na barra de inserir o nome, o teclado normal aparece, caso mude para a barra de inserir a idade, o teclado normal some e o teclado numérico é mostrado. As informações são guardadas em uma variável estática: o nome vai entrar na variável *NomeUsuario* e a idade entra na *IdadeUsuario*.

Figura 21: Cadastro com Teclado

Cadastro

Nome

Idade

Q W E R T Y U I O P

A S D F G H J K L Ç

↑ Z X C V B N M , .

Fonte: Autoria Própria

Figura 22: Cadastro com Teclado Numérico

Cadastro

Nome

Idade

7 8 9

4 5 6

1 2 3

0

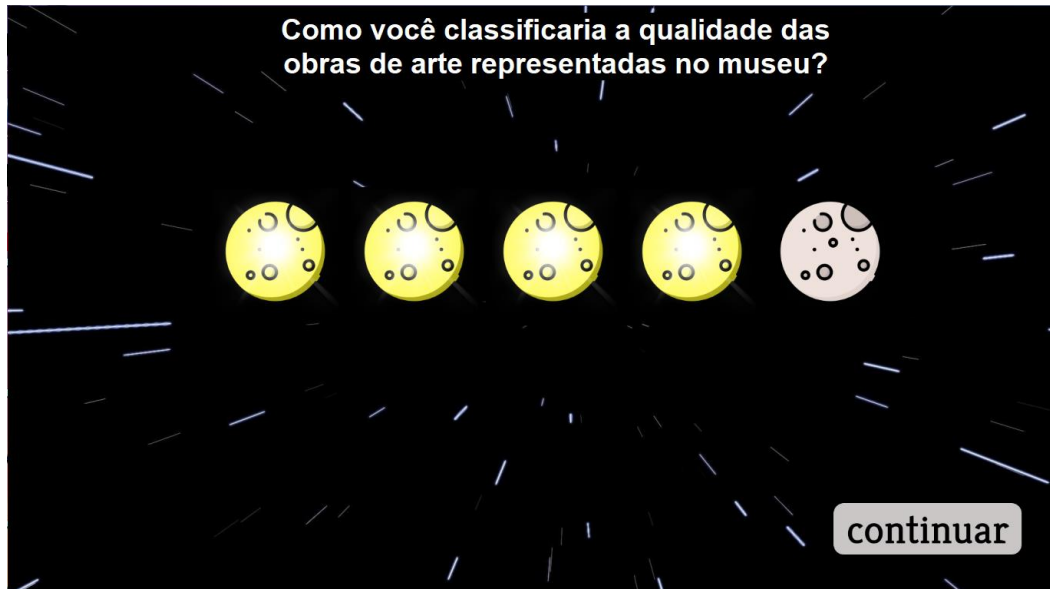
Fonte: Autoria Própria

Armazenando o nome e a idade do usuário, a tela muda para a janela *AvaliacaoMuseu*. Esta janela contém três perguntas relacionadas ao museu, que recebe nota de 1 a 5 usando luas como pode ser mostrado na figura 23, e um comentário opcional para o museu que é representado na figura 24.

Esta janela usa a mesma lógica da função *MudarObras* da lista de obras e da função *ProximaPergunta* do *Questionario*. Ao usuário avaliar o museu e deixar seu

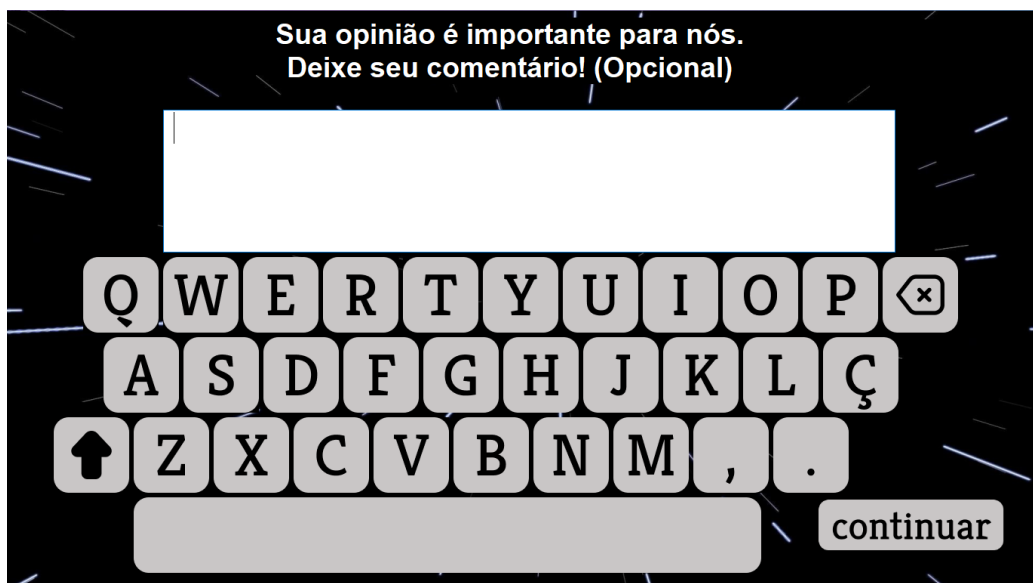
comentário com o teclado, pressionando o botão *Continuar*, a janela *AvaliacaoMuseu* chama a classe *Controle* para usar a função *ConcluirCadastro*, onde armazena as avaliações do museu e o comentário opcional nas variáveis estáticas. Após armazenar, o controle vai mandar um sinal para o *Form1 (Menu Principal)* com a mensagem “Armazenar”.

Figura 23: Avaliação Nota 1 a 5



Fonte: Autoria Própria

Figura 24: Avaliação comentário opcional



Fonte: Autoria Própria

As janelas *Questionario*, *CadastroUsuario* e *AvaliacaoMuseu* fecham e uma janela nova abre, chamada *Estatisticas*. A *Estatisticas* é uma janela onde mostra as respostas selecionadas, as respostas certas são exibidas com a fonte verde, ou caso o usuário tenha errado alguma, a resposta dele estará vermelha e riscada, e embaixo aparece a resposta certa, como está demonstrado na figura 25, após clicar no botão *Continuar*, é mostrado as estatísticas globais. A estatísticas globais contêm frases diferentes, três para caso o usuário acerte uma questão, e três para caso o usuário erre uma questão, estas frases são selecionadas de forma aleatória e mostrada para o usuário. Junto contém o total de erros e acertos de todas as perguntas juntas, como está na figura 26.

Figura 25: Estatísticas - Respostas do Usuário



Fonte: Autoridade Própria

Figura 26: Estatísticas - Estatística Global



Fonte: Autoridade Própria

Apertando o botão *Continuar*, é mostrado a tabela de liderança, contendo o nome, nota do questionário e a posição dos usuários que já passaram, como a figura 27 representa. Em seguida é mostrado a tela de agradecimento como está na figura 28, e o sistema volta para o *Menu Principal*

Figura 27: Tabela de Liderança

Tabela de Liderança		
Posicao	Nome	Nota
1	Waldir	5
2	Paula	5
3	Isac	4
4	Roberta	4
5	Sophia Morelli	4
6	Renata Assumpção	4
7	Leandro	4
8	Thais	3
9	Roger	3
10	Julia Pereira	2
11	Ana	1

continuar

Fonte: Autoria Própria

Figura 28: Tela de Agradecimento



Fonte: Autoria Própria

5.3.1 Janela Escondida

Como foi dito anteriormente na janela *AvaliacaoMuseu*, quando a função *ConcluirCadastro* da classe *Controle* é usado, o *Menu principal* recebe um sinal com a mensagem “Armazenar”. Antes das janelas fecharem, elas verificam se a variável *mensagemForm1* está com a mensagem “Armazenar”, caso esteja, a janela vai

armazenar esta mensagem e passar para a próxima janela até chegar no *Menu Principal*, como se fosse um telefone-sem-fio. Chegando no *Menu Principal*, o sistema chama a janela *GerenciadorUsuarios* que já é inicializado quando o sistema abre.

O *GerenciadorUsuarios* é uma janela que não pode ser vista pelos usuários, mas deve funcionar como segundo plano. A janela tem um estilo simples, somente com um texto informando que a janela não pode ser vista, e uma tabela que mostra os usuários que passaram pelo questionário, como pode ser visto na figura a seguir:

Figura 29: Gerenciador de Usuários

Esta Janela não podera aparecer
 Este é um gerenciador de Usuarios
 Este forms rodara de forma invisivel no sistema
 Quando o sistema for mostrar o relatorio ou a tabela de liderança, o Gerenciador de Usuarios entregará as informações

Esta janela não pode ser fechada, e só pode ser somente para fins de manutenção

Posicao	Nome	Idade	QuestionarioNota	Comentario
0	Waldir	98	5	Bom, isso me
1	Paula	42	5	Interessante
2	Isac	21	4	Achei Fraco, r
3	Roberta	20	4	Muito bom a
4	Sophia Morelli	15	4	Achei divertio
5	Renata Assump...	45	4	
6	Thais	25	3	LEGAL
7	Roger	33	3	Muito inform

Fonte: Autoria Própria

Quando o *Menu Principal* recebe a mensagem “Armazenar”, ele instrui o *GerenciadorUsuarios* a salvar os dados das variáveis estáticas em uma *List* que utiliza uma classe que contém propriedades (*Posicao*, *Nome*, *Idade*, *QuestionarioNota* e *Comentário*). Esse processo de armazenamento no *GerenciadorUsuarios* é importante, pois evita que informações de usuários que abandonam o questionário sejam salvas, garantindo a integridade dos relatórios.

Além disso, o *GerenciadorUsuarios* reorganiza os usuários salvos com base na propriedade *QuestionarioNota*, atribuindo valores à propriedade *Posicao* para exibição na Tabela de Liderança.

Depois de salvar o usuário na *List*, o *Form1* instrui o *GerenciadorUsuarios* a salvar a *List* em um arquivo *JSON*. Para criar ou modificar um arquivo *JSON*, é utilizado a biblioteca *Newtonsoft.Json*. O *GerenciadorUsuarios* passa o objeto da *List* para a função *SerializarLista* da classe *JsonControle*. Essa função converte o objeto para o formato *JSON* e salva no disco local (*C:/LIT/BancoDeDados*).

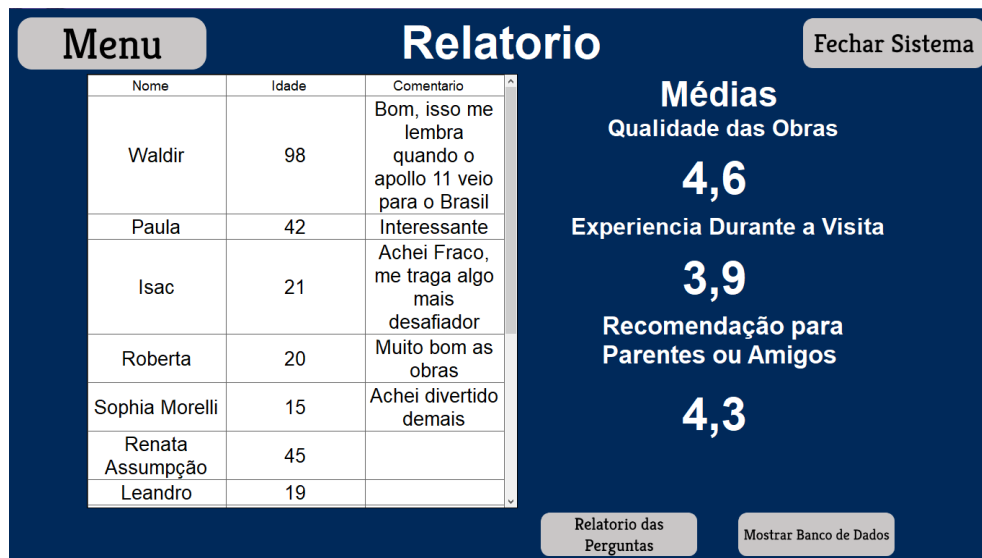
Após o *GerenciadorUsuarios* salvar o arquivo, tanto a Tabela de Liderança quanto o *Relatório* podem ler o arquivo para serem exibidos na tela, utilizando a função *DeserializarLista* da classe *JsonControle*.

5.3.2 Relatório

O sistema inclui uma funcionalidade de *Relatório*, onde é possível ver os feedbacks dos usuários e as avaliações do museu. O *Relatório* não pode ser uma opção de fácil acesso, para isso o *Relatório* só pode ser aberto se pressionar a tecla *CTRL* (*Control*). Como foi dito anteriormente, o dispositivo não possui um teclado físico, mas é possível inserir um usando uma entrada *USB* (*Universal Serial Bus*), que só os funcionários teriam acesso.

De acordo como a figura 30 demonstra, o *Relatório* exibe uma tabela dos usuários que finalizaram o questionário, a média das avaliações do museu, um botão para mostrar a pasta do banco de dados, um botão para abrir o relatório de perguntas e um botão para finalizar o sistema. Como foi falado anteriormente, o sistema não possui uma borda, para que os usuários não pudessem arrastar ou fechar o software, então a única forma de finalizar seria de um funcionário abrir o *Relatório* e apertar o botão *Fechar Sistema*

Figura 30: Relatório



Fonte: Autoria Própria

O *Relatório das Perguntas* demonstra quais perguntas os usuários estão tendo mais facilidade ou dificuldade, como a figura a seguir demonstra:

Figura 31: Relatório das Perguntas



Fonte: Autoria Própria

5.3.3 Teclado

A lógica do teclado no sistema do totem é simples e bem otimizada. Cada tecla possui dois IDs: o primeiro ID corresponde às letras minúsculas e o segundo às letras maiúsculas. Quando uma tecla é pressionada, a classe *Controle* é chamada e envia

o ID da tecla para um *switch*. O *switch* identifica a posição do ID e retorna o caractere correspondente, que é então inserido diretamente na *TextBox* selecionada. O *switch* abrange IDs para letras minúsculas, maiúsculas, números e teclas especiais.

Além disso, o teclado possui uma funcionalidade para alternar entre letras minúsculas e maiúsculas, bem como para inserir caracteres especiais e números, proporcionando uma experiência de digitação completa para o usuário.

5.3.4 Imagens

O sistema possui várias imagens que mudam conforme o usuário interage com o totem. Essas imagens são armazenadas na pasta do software, organizadas por categorias para facilitar a programação e manutenção. As categorias incluem imagens de botões, obras, enfeites, teclas e plano de fundo.

A organização cuidadosa das imagens permite que os desenvolvedores encontrem e atualizem facilmente os recursos visuais necessários, melhorando a eficiência do desenvolvimento e a experiência do usuário. As imagens são carregadas dinamicamente conforme necessário, garantindo que o sistema seja responsivo e ofereça uma navegação fluida.

5.3.5 Newtonsoft.Json

A serialização é o processo de converter objetos em uma representação *JSON*. Quando você serializa um objeto em C# usando o *Newtonsoft.Json*, você utiliza o método “*JsonConvert.SerializeObject(objeto)*” Este método recebe um objeto como entrada e retorna uma string *JSON* que representa esse objeto. Por exemplo quando o *GerenciadorUsuarios* vai serializar a *List* de usuários como mostra a imagem a seguir:

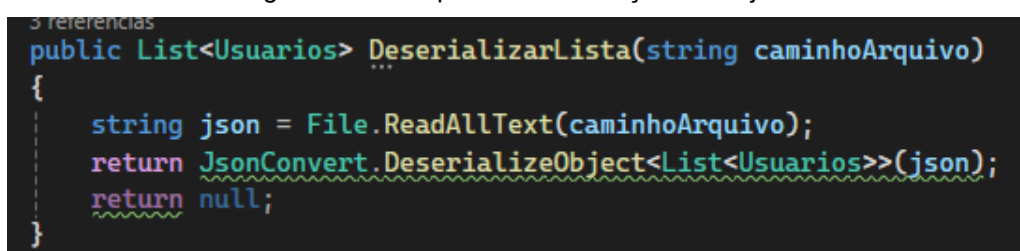
Figura 32: Exemplo - Serialização de Objeto

```
var strJson = JsonConvert.SerializeObject(usuarios, Formatting.Indented);
SalvarArquivo(strJson, caminhoArquivo);
```

Fonte: Autoria Própria

Por outro lado, a deserialização é o processo inverso, onde você converte uma *string JSON* de volta para um objeto C#. Para deserializar uma *string JSON* em um objeto em C#, você usa o método “*JsonConvert.DeserializeObject<T>(jsonString)*”. Este método recebe uma *string JSON* como entrada e retorna um objeto do tipo especificado *<T>*. Por exemplo quando o totem vai abrir o *Relatório* ou a *Tabela de Liderança*, onde o objeto do *<T>* é uma *List*, e a função retorna uma *List*, como está sendo demonstrado na figura a seguir:

Figura 33: Exemplo - Deserialização de Objeto



```

3 referencias
public List<Usuarios> DeserializarLista(string caminhoArquivo)
{
    string json = File.ReadAllText(caminhoArquivo);
    return JsonConvert.DeserializeObject<List<Usuarios>>(json);
    return null;
}

```

Fonte: Autoria Própria

O *Formatting.Indented* visto na figura 32 é um parâmetro usado no método “*JsonConvert.SerializeObject()*” que formata a saída *JSON* com indentação, tornando o *JSON* resultante mais legível. Isso significa que cada nível de aninhamento no *JSON* será recuado com espaços, facilitando a leitura e compreensão da estrutura do *JSON*.

5.3.6 Tempo de Ausência

A maioria das janelas desse sistema possui um temporizador que monitora a inatividade do usuário. Após um período predefinido de inatividade, o sistema fecha a janela automaticamente, voltando para o *Menu Principal*. Essa funcionalidade serve para otimizar o desempenho do sistema e evita que novos usuários encontrem janelas no meio do sistema, garantindo uma experiência de uso mais organizada e eficiente.

5.4 MANUAL DE DESINSTALAÇÃO

A desinstalação do sistema é tão simples quanto a instalação, bastando excluir a pasta “*SistemaTotem*”, que contém o arquivo executável e outros arquivos. Como mencionado anteriormente, o banco de dados não persiste no armazenamento, nesse

caso, apenas as pastas onde o banco de dados era criado permanecerão. Essas pastas não precisam ser excluídas, pois não contêm arquivos, ficando a critério do funcionário mantê-las ou excluí-las.

CONCLUSÃO

Este projeto teve como objetivo principal aprimorar a interatividade e o aprendizado nos museus. O sistema desenvolvido facilita a busca de informação do museu, fornecendo mapa, informações detalhadas sobre as obras em exibição e incentiva os visitantes a fazerem um questionário para competir com outros visitantes. Além da abordagem entreter o público, ela também promove o aprendizado do tema exposto.

Acreditamos que o resultado deste projeto se deve as disciplinas ministradas e ao esforço da Equipe LIT, pois com ele podemos utilizar e reutilizar o que foi criado neste projeto. Sem o banco de dados, a equipe não poderia fazer um banco de dados para armazenar os usuários. Sem Análise Orientada a Objeto, a equipe não conseguiria fazer um plano bem estruturado. Sem Engenharia de Software, o software não teria suas qualidades, e nossa equipe não se planejava em fazer algo reutilizável para projetos futuros. Sem Projeto de Interface com o Usuário, o nosso software não seria agradável para os usuários. Sem Programação Orientada a Objeto, o nosso software não seria de fácil manutenção e possivelmente não sairia do chão.

BIBLIOGRAFIA

ABNT - Associação Brasileira de Normas Técnicas. Engenharia de software – Qualidade de produto. Parte 1: Modelo de qualidade. Disponível em: **<https://jkolb.com.br/wp-content/uploads/2014/02/NBR-ISO_IEC-9126-1.pdf>**

Acesso em: 02.04.2024.

Cabos Patch Cord. Disponível em: **<https://www.cabospatchcord.com.br/patch-cord/cobre/cat6/cabo-de-rede-15-metros-patch-cord-cat6-utp-100-cobre?variant_id=1347>**. Acesso: 20/05/2024.

CIÊNCIA E CULTURA. **Chegada do Homem à Lua**. Vol.61 no.3 São Paulo, 2009.

FRIEDLAENDER, C. Espaço do conhecimento – UFMG. Disponível em: **<<https://www.ufmg.br/espacodoconhecimento/descubra-como-a-nasa-planeja-voltar-a-lua/>>**. Acesso em 22 de março de 2024.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002.

ISO/IEC 9126-1. 2001. **International Standard. Information Technology. Software engineering – Product quality – Part 2: External metrics**. 2001.

KOCHANSKI, DJONE. Engenharia de Software. Disponível em: **<<https://www.uniasselvi.com.br/extranet/layout/request/trilha/materiais/livro/livro.php?codigo=14602>>**. Acesso em 27 de março de 2024

MACEDO, R. T. FRANCISCATTO, R. CUNHA, G. B. BERTOLINI, C. **Licenciatura em Computação – Redes de Computadores**. Santa Maria-RS, 2018.

NASA. Gallery – Apollo 11. Disponível em: **<<https://www.nasa.gov/gallery/apollo-11/>>**. Acesso: 19/03/2024.

PEREIRA W. A gente precisa ver o Luar. In: COLONESE, P. (Org.) Os Mensageiros das Estrelas: Sistema Solar, volume 6, – Rio de Janeiro: Fiocruz – COC, 2021. P. 13-

33. Disponível em:
 <https://museudavida.fiocruz.br/images/Publicacoes_Educacao/PDFs/OMESSolar2021vol6.pdf>. Acesso em 22 de março. 2024.

PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software: Uma Abordagem Profissional. Porto Alegre: AMGH Editora Ltda, 9ª ed., 1996. Disponível em:
 <https://www.academia.edu/89376481/PRESSMAN_Engenharia_de_software_Uma_Abordagem_Profissional_9a_Ed>. Acesso em 20 mar. 2024.

PROJECT MANAGEMENT INSTITUTE (PMI). **Um Guia do Conhecimento em Gerenciamento de Projetos: Guia PMBOK**. Newtown Square, PA: PROJECT MANAGEMENT INSTITUTE, 2017.

SOMMERVILLE, IAN. Engenharia de Software. 2011. Disponível em: <<https://www.uniasselvi.com.br/extranet/layout/request/trilha/materiais/livro/livro.php?codigo=14602>>. Acesso em 27 de março de 2024.

Totem Digital Touch Screen 18,5 Polegadas IDX Smart – Index Soluções: Disponível em <<https://www.solucoesindex.com.br/totem-185-polegadas-touchscreen-sem-impressora>>. Acesso: 21/05/2024.

TEIXEIRA, Fabricio. **Introdução e boas práticas em UX Design**. [s.l.]: Editora Casa do Código, 2014.

TP-link. Disponível em: <<https://www.tp-link.com/br/home-networking/soho-switch/tl-sg108/>>. Acesso: 20/05/2024.

TP-link. Disponível em: <<https://www.tp-link.com/br/home-networking/wifi-router/archer-ax73/>>. Acesso: 20/05/2024.

APÊNDICE A - PERGUNTAS E RESPOSTAS

1 – Em que ano aconteceu a missão Apollo 11?

- A) 1959
- B) 1974
- C) 1965
- D) 1969

Resposta: D.

2 – Como é chamado o veículo de lançamento que aparece no quadro “Programa Saturno Apollo”?

- A) Apollo 11
- B) Foguete Espacial
- C) Saturn V
- D) Starship

Resposta: C.

3 – Quem foi o piloto do modulo lunar da missão de pouso lunar Apollo 11?

- A) Edwin E. Aldrin Jr.
- B) Neil Armstrong
- C) Elton John
- D) Michael Collins

Resposta: A.

4 – O que é EASEP?

- A) Early Apollo Satisfying Experimentation Pocket

- B) Early Apollo Scientific Experiments Package
- C) Eletronic Art Sports Equestrian Paintball
- D) Earth Air Salt Earn Pearls

Resposta: B.

5 – Quem renunciou o programa espacial da Nasa?

- A) Neil Armstrong
- B) Michael Jordan
- C) Michael Collins
- D) Edwin E. Aldrin Jr.

Resposta: C

APÊNDICE B – FORM1 (MENU PRINCIPAL)

```

using Microsoft.VisualBasic.Devices;
using Microsoft.VisualBasic.FileIO;
using SistemaTotemHomemALua.Modelos;

namespace SistemaTotemHomemALua
{
    public partial class Form1 : Form
    {
        public GerenciadorUsuarios gerUsuarios;
        public Relatorio rel;
        public Form1()
        {
            InitializeComponent();
            gerUsuarios = new GerenciadorUsuarios();
            gerUsuarios.Show();
            gerUsuarios.Visible = false;
        }

        string caminhoDiretorio = @"C:\LIT\BancoDeDados";
        string caminhoArquivo = @"C:\LIT\BancoDeDados\usuarios.json";

        private void pcbObras_Click(object sender, EventArgs e)
        {
            ListaObras listaObras = new ListaObras();
            listaObras.ShowDialog();
            this.Visible = true;
        }

        private void pcbQuestionario_Click(object sender, EventArgs e)
        {
            Questionario questionario = new Questionario();
            questionario.ShowDialog();

            if (questionario.mensagemForm1.Equals("Armazenar"))
            {
                gerUsuarios.AdicionarPessoa(questionario.mensagemForm1,
                    Estatico.NOMEUSUARIO, Estatico.IDADEUSUARIO, Estatico.QUESTIONARIOUOTAUSUARIO,
                    Estatico.COMENTARIOUSUARIO);
                gerUsuarios.SalvarSerializar(caminhoDiretorio, caminhoArquivo);
                Estatisticas estatisticas = new Estatisticas();
                estatisticas.ShowDialog();
                TabelaDeLideranca tabLideranca = new
                TabelaDeLideranca(caminhoArquivo);
                tabLideranca.ShowDialog();
            }

            this.Visible = true;
        }

        private void pcbMapa_Click(object sender, EventArgs e)
        {
            Mapa mapa = new Mapa();
            mapa.ShowDialog();
            this.Visible = true;
        }

        private void Form1_KeyDown(object sender, KeyEventArgs e)
        {

```

```

        if (e.KeyCode == Keys.ControlKey)
        {
            rel = new Relatorio(caminhoArquivo);
            rel.ShowDialog();
            if (rel.mensagemForm1.Equals("FinalizarSistema"))
            {
                this.Close();
            }
        }
    }

    private void pcbPreencherTabela_Click(object sender, EventArgs e)
    {
        gerUsuarios.AdicionarPessoaPreencherTabela("Isac", 21, 4, "Achei
        Fraco, me traga algo mais desafiador");
        gerUsuarios.AdicionarPessoaPreencherTabela("Thais", 25, 3, "LEGAL");
        gerUsuarios.AdicionarPessoaPreencherTabela("Roberta", 20, 4, "Muito
        bom as obras");
        gerUsuarios.AdicionarPessoaPreencherTabela("Waldir", 98, 5, "Bom,
        isso me lembra quando o apollo 11 veio para o Brasil");
        gerUsuarios.AdicionarPessoaPreencherTabela("Sophia Morelli", 15, 4,
        "Achei divertido demais");
        gerUsuarios.AdicionarPessoaPreencherTabela("Roger", 33, 3, "Muito
        informativo");
        gerUsuarios.AdicionarPessoaPreencherTabela("Julia Pereira", 16, 2,
        "Tipo, achei legal sabe");
        gerUsuarios.AdicionarPessoaPreencherTabela("Luiza", 14, 0, "");
        gerUsuarios.AdicionarPessoaPreencherTabela("Ana", 17, 1, "");
        gerUsuarios.AdicionarPessoaPreencherTabela("Richard", 20, 1, "Achei
        as perguntas dificil demais");
        gerUsuarios.AdicionarPessoaPreencherTabela("Sofia Guedes", 15, 0,
        "");
        gerUsuarios.AdicionarPessoaPreencherTabela("Paula", 42, 5,
        "Interessante");
        gerUsuarios.AdicionarPessoaPreencherTabela("Renata Assumpção", 45, 4,
        "");

        Estatico.SOMATOTALQUALIDADEOBRAS += 64;
        Estatico.NUMEROAVALIACOESQUALIDADEOBRAS += 14;

        Estatico.SOMATOTALVISITAMUSEU += 57;
        Estatico.NUMEROAVALIACOESVISITAMUSEU += 14;

        Estatico.SOMATOTALRECOMENDACAO = 61;
        Estatico.NUMEROAVALIACOESRECOMENDACAO += 14;

        Estatico.ACERTOSQUEST1 += 5;
        Estatico.ACERTOSQUEST2 += 9;
        Estatico.ACERTOSQUEST3 += 4;
        Estatico.ACERTOSQUEST4 += 8;
        Estatico.ACERTOSQUEST5 += 8;

        Estatico.ERROSQUEST1 += 9;
        Estatico.ERROSQUEST2 += 5;
        Estatico.ERROSQUEST3 += 10;
        Estatico.ERROSQUEST4 += 6;
        Estatico.ERROSQUEST5 += 6;

        Estatico.TOTALRESPOSTASQUEST1 += Estatico.ACERTOSQUEST1 +
        Estatico.ERROSQUEST1;
        Estatico.TOTALRESPOSTASQUEST2 += Estatico.ACERTOSQUEST2 +
        Estatico.ERROSQUEST2;
    }

```

```

        Estatico.TOTALRESPOSTASQUEST3 += Estatico.ACERTOSQUEST3 +
Estatico.ERROSQUEST3;
        Estatico.TOTALRESPOSTASQUEST4 += Estatico.ACERTOSQUEST4 +
Estatico.ERROSQUEST4;
        Estatico.TOTALRESPOSTASQUEST5 += Estatico.ACERTOSQUEST5 +
Estatico.ERROSQUEST5;

        Estatico.TOTALERROS = Estatico.ERROSQUEST1 + Estatico.ERROSQUEST2 +
Estatico.ERROSQUEST3 + Estatico.ERROSQUEST4 + Estatico.ERROSQUEST5;
        Estatico.TOTALACERTOS = Estatico.ACERTOSQUEST1 +
Estatico.ACERTOSQUEST2 + Estatico.ACERTOSQUEST3 + Estatico.ACERTOSQUEST4 +
Estatico.ACERTOSQUEST5;

        gerUsuarios.SalvarSerializar(caminhoDiretorio, caminhoArquivo);

        pcbPreencherTabela.Visible = false;
    }

    private void Form1_FormClosed(object sender, FormClosedEventArgs e)
    {
        File.Delete(caminhoArquivo);
    }

}
}

```

APÊNDICE C – MAPA

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaTotemHomemALua
{
    public partial class Mapa : Form
    {
        public Mapa()
        {
            InitializeComponent();
        }

        private void pbxBotaoMenu_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void tmrTempoAusencia_Tick(object sender, EventArgs e)
        {
            this.Close();
        }

        public void ResetTimer()
        {
            tmrTempoAusencia.Enabled = false;
            tmrTempoAusencia.Enabled = true;
        }

        private void pcbMapa_Click(object sender, EventArgs e)
        {
            ResetTimer();
        }

        private void Mapa_Click(object sender, EventArgs e)
        {
            ResetTimer();
        }
    }
}

```

APÊNDICE D – LISTA OBRAS

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaTotemHomemALua
{
    public partial class ListaObras : Form
    {
        public ListaObras()
        {
            InitializeComponent();
            MudarObras();
        }

        int paginaID = 1;
        int idQuadro1 = 1;
        int idQuadro2 = 2;

        private void pcbQuadroId1_Click(object sender, EventArgs e)
        {
            tmrTempoAusencia.Enabled = false;
            ObrasDescricoes obrasDescricoes = new ObrasDescricoes(idQuadro1);
            obrasDescricoes.ShowDialog();
            tmrTempoAusencia.Enabled = true;
        }

        private void pcbQuadroId2_Click(object sender, EventArgs e)
        {
            tmrTempoAusencia.Enabled = false;
            ObrasDescricoes obrasDescricoes = new ObrasDescricoes(idQuadro2);
            obrasDescricoes.ShowDialog();
            tmrTempoAusencia.Enabled = true;
        }

        private void pbxBotaoMenu_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        public void ResetTimer()
        {
            tmrTempoAusencia.Enabled = false;
            tmrTempoAusencia.Enabled = true;
        }

        public void MudarPagina()
        {
            switch (paginaID)
            {
                case 1:

```

```
idQuadro1 = 1;
idQuadro2 = 2;

MudarObras();

pbxBotaoPaginaUp.Visible = false;
pbxBotaoPaginaDown.Visible = true;

break;
case 2:

idQuadro1 = 3;
idQuadro2 = 4;

MudarObras();

pbxBotaoPaginaUp.Visible = true;
pbxBotaoPaginaDown.Visible = true;

break;
case 3:

idQuadro1 = 5;
idQuadro2 = 6;

MudarObras();

pbxBotaoPaginaUp.Visible = true;
pbxBotaoPaginaDown.Visible = true;

break;
case 4:

idQuadro1 = 7;
idQuadro2 = 8;
MudarObras();

pbxBotaoPaginaUp.Visible = true;
pbxBotaoPaginaDown.Visible = true;
break;
case 5:

idQuadro1 = 9;
idQuadro2 = 10;
MudarObras();

pbxBotaoPaginaUp.Visible = true;
pbxBotaoPaginaDown.Visible = false;
break;
default:

paginaID = 1;
MudarPagina();
```

```

        MudarObras();
        MessageBox.Show("Não tem mais pagina, voltando para a pagina
1");
        break;
    }
}

public void MudarObras()
{
    switch (paginaID)
    {
        case 1:
            pcbQuadroId1.ImageLocation = "Imagens/Obras/Astronauta Neil
Armstrong.jpg";
            pcbQuadroId2.ImageLocation = "Imagens/Obras/Astronauta Edwin
E. Aldrin Jr.jpg";

            lblQuadroId1.Text = "Astronauta Neil Armstrong";
            lblQuadroId2.Text = "Astronauta Edwin E. Aldrin Jr";
            break;

        case 2:
            pcbQuadroId1.ImageLocation = "Imagens/Obras/Astronauta
Michael Collins.jpg";
            pcbQuadroId2.ImageLocation = "Imagens/Obras/A tripulação da
Apollo 11.jpg";

            lblQuadroId1.Text = "Astronauta Michael\nCollins";
            lblQuadroId2.Text = "A tripulação da Apollo 11";
            break;

        case 3:
            pcbQuadroId1.ImageLocation = "Imagens/Obras/Saturn V
Ventures.jpg";
            pcbQuadroId2.ImageLocation = "Imagens/Obras/Programa Saturno
Apollo.jpg";

            lblQuadroId1.Text = "Saturn V Venture";
            lblQuadroId2.Text = "Programa Saturno Apollo";
            break;

        case 4:
            pcbQuadroId1.ImageLocation = "Imagens/Obras/Módulo Lunar
Apollo.jpg";
            pcbQuadroId2.ImageLocation = "Imagens/Obras/Módulo Lunar da
Apollo 11 na superfície lunar.jpg";

            lblQuadroId1.Text = "Módulo Lunar Apollo";
            lblQuadroId2.Text = "Módulo Lunar da Apollo 11\nna superfice
lunar";
            break;

        case 5:
            pcbQuadroId1.ImageLocation = "Imagens/Obras/Pegada no solo
lunar.jpg";

```

```

        lua.jpg";

        pcbQuadroId2.ImageLocation = "Imagens/Obras/Vista da

        lblQuadroId1.Text = "Pegada no solo lunar";
        lblQuadroId2.Text = "Vista da lua";
        break;

    default:

        break;

    }

}

private void pbxBotaoPaginaUp_Click(object sender, EventArgs e)
{

    ResetTimer();

    paginaID -= 1;
    MudarPagina();
}

private void pbxBotaoPaginaDown_Click(object sender, EventArgs e)
{

    ResetTimer();

    paginaID += 1;
    MudarPagina();

}

private void ListaObras_Click(object sender, EventArgs e)
{

    ResetTimer();

}

private void tmrTempoAusencia_Tick(object sender, EventArgs e)
{

    this.Close();

}

}
}

```


APÊNDICE E – OBRASDESCRICOES

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaTotemHomemALua
{
    public partial class ObrasDescricoes : Form
    {
        public ObrasDescricoes(int id)
        {
            InitializeComponent();
            AtualizarTela(id);
        }

        public void ResetTimer()
        {
            tmrTempoAusencia.Enabled = false;
            tmrTempoAusencia.Enabled = true;
        }

        public void AtualizarTela(int id)
        {
            switch (id)
            {
                case 1:
                    pcbObraImagem.ImageLocation = "Imagens/Obras/Astronauta Neil
Armstrong.jpg";
                    lblObraTitulo.Text = "Astronauta Neil Armstrong";
                    lblObraDescricao.Text = "Neil Armstrong, vestido com seu
traje espacial\nnpose para seu retrato oficial da Apollo 11.";
                    break;

                case 2:
                    pcbObraImagem.ImageLocation = "Imagens/Obras/Astronauta Edwin
E. Aldrin Jr.jpg";
                    lblObraTitulo.Text = "Astronauta Edwin E. Aldrin Jr";
                    lblObraDescricao.Text = "Astronauta Edwin E. Aldrin Jr.
Aldrin\nfoi piloto do módulo lunar da missão de pouso lunar Apollo 11";
                    break;

                case 3:
                    pcbObraImagem.ImageLocation = "Imagens/Obras/Astronauta
Michael Collins.jpg";
                    lblObraTitulo.Text = "Astronauta Michael Collins";
                    lblObraDescricao.Text = "Collins renunciou ao programa
espacial da NASA\ndesde que esta foto foi tirada";
                    break;

                case 4:
                    pcbObraImagem.ImageLocation = "Imagens/Obras/A tripulação da
Apollo 11.jpg";
                    lblObraTitulo.Text = "A tripulação da Apollo 11";

```

```

        lblObraDescricao.Text = "A tripulação da Apollo 11 realiza
uma verificação funcional\ne de ajuste do compartimento da tripulação, dos
equipamentos\ne locais de armazenamento, em seu módulo de comando";
        break;

    case 5:
        pcbObraImagem.ImageLocation = "Imagens/Obras/Saturn V
Ventures.jpg";
        lblObraTitulo.Text = "Saturn V Venture";
        lblObraDescricao.Text = "Carregando o foguete Apollo 11
Saturn V e o lançador móvel\no rastreador avança lentamente ao longo da jornada
de cinco quilômetros\ntaté a plataforma de lançamento 39A";
        break;

    case 6:
        pcbObraImagem.ImageLocation = "Imagens/Obras/Programa Saturno
Apollo.jpg";
        lblObraTitulo.Text = "Programa Saturno Apollo";
        lblObraDescricao.Text = "Esta fotografia mostra o veículo de
lançamento Saturn V (SA-506)\npara a decolagem da missão Apollo 11 às 8h32 CDT,
16 de julho de 1969\ndo complexo de lançamento 39A no Centro Espacial Kennedy";
        break;

    case 7:
        pcbObraImagem.ImageLocation = "Imagens/Obras/Módulo Lunar
Apollo.jpg";
        lblObraTitulo.Text = "Módulo Lunar Apollo";
        lblObraDescricao.Text = "O estágio de subida do Módulo Lunar
da Apollo 11, com os astronautas\nNeil A. Armstrong e Edwin E. Aldrin Jr. a
bordo\né fotografado dos Módulos de Comando e Serviço (CSM)\ndurante o encontro
na órbita lunar";
        break;

    case 8:
        pcbObraImagem.ImageLocation = "Imagens/Obras/Módulo Lunar da
Apollo 11 na superfície lunar.jpg";
        lblObraTitulo.Text = "Módulo Lunar da Apollo 11 na superfice
lunar";
        lblObraDescricao.Text = "O astronauta Edwin E. Aldrin Jr.,
piloto do módulo lunar\nse prepara para implantar o\nEarly Apollo Scientific
Experiments Package (EASEP)\ndurante a atividade extraveicular da superfície
lunar (EVA) da Apollo 11";
        break;

    case 9:
        pcbObraImagem.ImageLocation = "Imagens/Obras/Pegada no solo
lunar.jpg";
        lblObraTitulo.Text = "Pegada no solo lunar";
        lblObraDescricao.Text = "Uma visão aproximada da bota e da
pegada de um astronauta no solo lunar\nfotografada com uma câmera de superfície
lunar de 70 mm\ndurante a atividade extraveicular da superfície lunar (EVA) da
Apollo 11";
        break;

    case 10:
        pcbObraImagem.ImageLocation = "Imagens/Obras/Vista da
lua.jpg";
        lblObraTitulo.Text = "Vista da lua";
        lblObraDescricao.Text = "Esta vista da espaçonave Apollo 11
mostra a Terra elevando-se\nacima do horizonte da lua. O terreno lunar retratado
está na área do\nMar de Smyth, no lado mais próximo";
        break;

```

```

        default:
            MessageBox.Show("Nenhuma Obra Selecionada, Selecione uma!!");
            this.Close();
            break;
    }

}

private void pbxBotaoVoltar_Click(object sender, EventArgs e)
{
    this.Close();
}

private void tmrTempoAusencia_Tick(object sender, EventArgs e)
{
    this.Close();
}

private void ObrasDescricoes_Click(object sender, EventArgs e)
{
    ResetTimer();
}

private void lblObraDescricao_Click(object sender, EventArgs e)
{
    ResetTimer();
}

private void pcbObraImagem_Click(object sender, EventArgs e)
{
    ResetTimer();
}

private void lblObraTitulo_Click(object sender, EventArgs e)
{
    ResetTimer();
}
}
}

```

APÊNDICE F – QUESTIONÁRIO

```

using SistemaTotemHomemALua.Modelos;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaTotemHomemALua
{
    public partial class Questionario : Form
    {
        public Questionario()
        {
            InitializeComponent();
            Estatico.QUESTIONARIOUOTAUSUARIO = 0;

            Estatico.TEMPACERTOSQUEST1 = 0;
            Estatico.TEMPACERTOSQUEST2 = 0;
            Estatico.TEMPACERTOSQUEST3 = 0;
            Estatico.TEMPACERTOSQUEST4 = 0;
            Estatico.TEMPACERTOSQUEST5 = 0;

            Estatico.TEMPERROSQUEST1 = 0;
            Estatico.TEMPERROSQUEST2 = 0;
            Estatico.TEMPERROSQUEST3 = 0;
            Estatico.TEMPERROSQUEST4 = 0;
            Estatico.TEMPERROSQUEST5 = 0;

            Estatico.TEMPRESPOSTAQUEST1 = "";
            Estatico.TEMPRESPOSTAQUEST2 = "";
            Estatico.TEMPRESPOSTAQUEST3 = "";
            Estatico.TEMPRESPOSTAQUEST4 = "";
            Estatico.TEMPRESPOSTAQUEST5 = "";

            ProximaPergunta(1);
        }

        public string mensagemForm1 = "";
        int respostaSelecionada = 0;
        int idPergunta = 1;
        string alternativaEscolhida = "";

        private void pbxBotaoMenu_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        public void ProximaPergunta(int idPergunta)
        {
            switch (idPergunta)
            {
                case 1:
                    respostaSelecionada = 0;

                    pcbAlternativaA.ImageLocation = "Imagens/Botões/BotaoA.png";
                    pcbAlternativaB.ImageLocation = "Imagens/Botões/BotaoB.png";

```

```

pcbAlternativaC.ImageLocation = "Imagens/Botões/BotaoC.png";
pcbAlternativaD.ImageLocation = "Imagens/Botões/BotaoD.png";

11?";
    lblPergunta.Text = "1 - Em que ano aconteceu a missão\nApollo

    lblAlternativaA.Text = "A) 1959";
    lblAlternativaB.Text = "B) 1974";
    lblAlternativaC.Text = "C) 1965";
    lblAlternativaD.Text = "D) 1969";

    lblMensagem.Visible = false;
    lblMensagem.Text = "";
    break;

case 2:
    respostaSelecionada = 0;

    pcbAlternativaA.ImageLocation = "Imagens/Botões/BotaoA.png";
    pcbAlternativaB.ImageLocation = "Imagens/Botões/BotaoB.png";
    pcbAlternativaC.ImageLocation = "Imagens/Botões/BotaoC.png";
    pcbAlternativaD.ImageLocation = "Imagens/Botões/BotaoD.png";

    lblPergunta.Text = "2 - Como é chamado o veículo de
lançamento\nque aparece no quadro\n“Programa Sa-turno Apollo”?";
    lblAlternativaA.Text = "A) Apollo 11";
    lblAlternativaB.Text = "B) Foguete Espacial";
    lblAlternativaC.Text = "C) Saturn V";
    lblAlternativaD.Text = "D) Starship";

    lblMensagem.Visible = false;
    lblMensagem.Text = "";
    break;

case 3:
    respostaSelecionada = 0;

    pcbAlternativaA.ImageLocation = "Imagens/Botões/BotaoA.png";
    pcbAlternativaB.ImageLocation = "Imagens/Botões/BotaoB.png";
    pcbAlternativaC.ImageLocation = "Imagens/Botões/BotaoC.png";
    pcbAlternativaD.ImageLocation = "Imagens/Botões/BotaoD.png";

    lblPergunta.Text = "3 - Quem foi o piloto do modulo lunar\nda
missão de pouso lunar Apollo 11?";
    lblAlternativaA.Text = "A) Edwin E. Aldrin Jr.";
    lblAlternativaB.Text = "B) Neil Armstrong";
    lblAlternativaC.Text = "C) Elton John";
    lblAlternativaD.Text = "D) Michael Collins";

    lblMensagem.Visible = false;
    lblMensagem.Text = "";
    break;

case 4:
    respostaSelecionada = 0;

    pcbAlternativaA.ImageLocation = "Imagens/Botões/BotaoA.png";
    pcbAlternativaB.ImageLocation = "Imagens/Botões/BotaoB.png";
    pcbAlternativaC.ImageLocation = "Imagens/Botões/BotaoC.png";
    pcbAlternativaD.ImageLocation = "Imagens/Botões/BotaoD.png";

    lblPergunta.Text = "4 - O que é EASEP?";
    lblAlternativaA.Text = "A) Early Apollo Satisfying
Experimentation Pocket";

```

```

        lblAlternativaB.Text = "B) Early Apollo Scientific
Experiments Package";
        lblAlternativaC.Text = "C) Eletronic Art Sports Equestrian
Paintball";
        lblAlternativaD.Text = "D) Earth Air Salt Earn Pearls";

        lblMensagem.Visible = false;
        lblMensagem.Text = "";
        break;

    case 5:
        respostaSelecionada = 0;

        pcbAlternativaA.ImageLocation = "Imagens/Botões/BotaoA.png";
        pcbAlternativaB.ImageLocation = "Imagens/Botões/BotaoB.png";
        pcbAlternativaC.ImageLocation = "Imagens/Botões/BotaoC.png";
        pcbAlternativaD.ImageLocation = "Imagens/Botões/BotaoD.png";

        lblPergunta.Text = "5 - Quem renunciou o programa\nespacial
da Nasa? ";

        lblAlternativaA.Text = "A) Neil Armstrong";
        lblAlternativaB.Text = "B) Michael Jordan";
        lblAlternativaC.Text = "C) Michael Collins";
        lblAlternativaD.Text = "D) Edwin E. Aldrin Jr.";

        lblMensagem.Visible = false;
        lblMensagem.Text = "";
        break;

    case 6:
        tmrTempoAusencia.Enabled = false;
        CadastroUsuario cadUsuario = new CadastroUsuario();
        cadUsuario.ShowDialog();
        if (cadUsuario.mensagemForm1.Equals("Armazenar"))
        {
            mensagemForm1 = cadUsuario.mensagemForm1;
        }
        this.Close();
        break;

    default:
        this.Close();
        break;
}

}

private void pcbAlternativaA_Click(object sender, EventArgs e)
{
    ResetTimer();
    respostaSelecionada = 1;    //Feito para não ocorrer um clique duplo
    alternativaEscolhida = lblAlternativaA.Text;
    pcbAlternativaA.ImageLocation = "Imagens/Botões/BotaoACheck.png";
    pcbAlternativaB.ImageLocation = "Imagens/Botões/BotaoB.png";
    pcbAlternativaC.ImageLocation = "Imagens/Botões/BotaoC.png";
    pcbAlternativaD.ImageLocation = "Imagens/Botões/BotaoD.png";
}

private void lblAlternativaA_Click(object sender, EventArgs e)
{
    ResetTimer();
    respostaSelecionada = 1;    //Feito para não ocorrer um clique duplo
    alternativaEscolhida = lblAlternativaA.Text;

```

```

        pcbAlternativaA.ImageLocation = "Imagens/Botões/BotaoACheck.png";
        pcbAlternativaB.ImageLocation = "Imagens/Botões/BotaoB.png";
        pcbAlternativaC.ImageLocation = "Imagens/Botões/BotaoC.png";
        pcbAlternativaD.ImageLocation = "Imagens/Botões/BotaoD.png";
    }

    private void pcbAlternativaB_Click(object sender, EventArgs e)
    {
        ResetTimer();
        respostaSelecionada = 1;    //Feito para não ocorrer um clique duplo
        alternativaEscolhida = lblAlternativaB.Text;
        pcbAlternativaA.ImageLocation = "Imagens/Botões/BotaoA.png";
        pcbAlternativaB.ImageLocation = "Imagens/Botões/BotaoBCheck.png";
        pcbAlternativaC.ImageLocation = "Imagens/Botões/BotaoC.png";
        pcbAlternativaD.ImageLocation = "Imagens/Botões/BotaoD.png";
    }

    private void lblAlternativaB_Click(object sender, EventArgs e)
    {
        ResetTimer();
        respostaSelecionada = 1;    //Feito para não ocorrer um clique duplo
        alternativaEscolhida = lblAlternativaB.Text;
        pcbAlternativaA.ImageLocation = "Imagens/Botões/BotaoA.png";
        pcbAlternativaB.ImageLocation = "Imagens/Botões/BotaoBCheck.png";
        pcbAlternativaC.ImageLocation = "Imagens/Botões/BotaoC.png";
        pcbAlternativaD.ImageLocation = "Imagens/Botões/BotaoD.png";
    }

    private void pcbAlternativaC_Click(object sender, EventArgs e)
    {
        ResetTimer();
        respostaSelecionada = 1;    //Feito para não ocorrer um clique duplo
        alternativaEscolhida = lblAlternativaC.Text;
        pcbAlternativaA.ImageLocation = "Imagens/Botões/BotaoA.png";
        pcbAlternativaB.ImageLocation = "Imagens/Botões/BotaoB.png";
        pcbAlternativaC.ImageLocation = "Imagens/Botões/BotaoCCheck.png";
        pcbAlternativaD.ImageLocation = "Imagens/Botões/BotaoD.png";
    }

    private void lblAlternativaC_Click(object sender, EventArgs e)
    {
        ResetTimer();
        respostaSelecionada = 1;    //Feito para não ocorrer um clique duplo
        alternativaEscolhida = lblAlternativaC.Text;
        pcbAlternativaA.ImageLocation = "Imagens/Botões/BotaoA.png";
        pcbAlternativaB.ImageLocation = "Imagens/Botões/BotaoB.png";
        pcbAlternativaC.ImageLocation = "Imagens/Botões/BotaoCCheck.png";
        pcbAlternativaD.ImageLocation = "Imagens/Botões/BotaoD.png";
    }

    private void pcbAlternativaD_Click(object sender, EventArgs e)
    {
        ResetTimer();
        respostaSelecionada = 1;    //Feito para não ocorrer um clique duplo
        alternativaEscolhida = lblAlternativaD.Text;
        pcbAlternativaA.ImageLocation = "Imagens/Botões/BotaoA.png";
        pcbAlternativaB.ImageLocation = "Imagens/Botões/BotaoB.png";
        pcbAlternativaC.ImageLocation = "Imagens/Botões/BotaoC.png";
        pcbAlternativaD.ImageLocation = "Imagens/Botões/BotaoDCheck.png";
    }

    private void lblAlternativaD_Click(object sender, EventArgs e)
    {
        ResetTimer();
    }

```

```

        respostaSelecionada = 1;    //Feito para não ocorrer um clique duplo
        alternativaEscolhida = lblAlternativaD.Text;
        pcbAlternativaA.ImageLocation = "Imagens/Botões/BotaoA.png";
        pcbAlternativaB.ImageLocation = "Imagens/Botões/BotaoB.png";
        pcbAlternativaC.ImageLocation = "Imagens/Botões/BotaoC.png";
        pcbAlternativaD.ImageLocation = "Imagens/Botões/BotaoDCheck.png";
    }

```

```

private void pbxBotaoContinuar_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (!respostaSelecionada.Equals(0))
    {
        Controle controle = new Controle();
        controle.ValidarResposta(idPergunta, alternativaEscolhida);
        idPergunta += 1;
        ProximaPergunta(idPergunta);
    }
    else
    {
        lblMensagem.Visible = true;
        lblMensagem.Text = "Selecione uma resposta!";
    }
}

```

```

public void ResetTimer()
{
    tmrTempoAusencia.Enabled = false;
    tmrTempoAusencia.Enabled = true;
}

```

```

private void tmrTempoAusencia_Tick(object sender, EventArgs e)
{
    this.Close();
}

```

```

private void Questionario_Click(object sender, EventArgs e)
{
    ResetTimer();
}

```

```

    }
}

```


APÊNDICE G – CADASTROUSUARIO

```

using SistemaTotemHomemALua.Modelos;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaTotemHomemALua
{
    public partial class CadastroUsuario : Form
    {
        public CadastroUsuario()
        {
            InitializeComponent();
            Estatico.NOMEUSUARIO = "";
            Estatico.IDADEUSUARIO = 0;
            lblMensagem.Visible = false;
            shiftOpcao = 1;
            AtualizarTeclas(shiftOpcao);

            //Para não começar com os dois teclados ativos
            pcbKeyA.Visible = false;
            pcbKeyB.Visible = false;
            pcbKeyC.Visible = false;
            pcbKeyD.Visible = false;
            pcbKeyE.Visible = false;
            pcbKeyF.Visible = false;
            pcbKeyG.Visible = false;
            pcbKeyH.Visible = false;
            pcbKeyI.Visible = false;
            pcbKeyJ.Visible = false;
            pcbKeyK.Visible = false;
            pcbKeyL.Visible = false;
            pcbKeyM.Visible = false;
            pcbKeyN.Visible = false;
            pcbKeyO.Visible = false;
            pcbKeyP.Visible = false;
            pcbKeyQ.Visible = false;
            pcbKeyR.Visible = false;
            pcbKeyS.Visible = false;
            pcbKeyT.Visible = false;
            pcbKeyU.Visible = false;
            pcbKeyV.Visible = false;
            pcbKeyW.Visible = false;
            pcbKeyX.Visible = false;
            pcbKeyY.Visible = false;
            pcbKeyZ.Visible = false;
            pcbKeyÇ.Visible = false;
            pcbKeyShift.Visible = false;
            pcbKeySpace.Visible = false;
            pcbKeyBackspace.Visible = false;
            pcbKeyComma.Visible = false;
            pcbKeyDot.Visible = false;

            pcbNumpad0.Visible = false;
            pcbNumpad1.Visible = false;
            pcbNumpad2.Visible = false;
        }
    }
}

```

```

        pcbNumpad3.Visible = false;
        pcbNumpad4.Visible = false;
        pcbNumpad5.Visible = false;
        pcbNumpad6.Visible = false;
        pcbNumpad7.Visible = false;
        pcbNumpad8.Visible = false;
        pcbNumpad9.Visible = false;
        pcbNumpadBackspace.Visible = false;
    }

    public string mensagemForm1 = "";
    int shiftOpcao = 1;

    private void pbxBotaoContinuar_Click(object sender, EventArgs e)
    {
        ResetTimer();
        Controle controle = new Controle();
        controle.ArmazenarNomeIdade(txbNome.Text, txbIdade.Text);
        if (!controle.Mensagem.Equals(""))
        {
            lblMensagem.Visible = true;
            lblMensagem.Text = controle.Mensagem;
        }
        else
        {
            tmrTempoAusencia.Enabled = false;
            AvaliacaoMuseu avaMuseu = new AvaliacaoMuseu();
            avaMuseu.ShowDialog();
            if (avaMuseu.mensagemForm1.Equals("Armazenar"))
            {
                mensagemForm1 = avaMuseu.mensagemForm1;
            }
            this.Close();
        }
    }

    #region Reset Timer Ausencia

    public void ResetTimer()
    {
        tmrTempoAusencia.Enabled = false;
        tmrTempoAusencia.Enabled = true;
    }

    private void tmrTempoAusencia_Tick(object sender, EventArgs e)
    {
        this.Close();
    }

    private void txbIdade_TextChanged(object sender, EventArgs e)
    {
        ResetTimer();
    }

    private void lblIdade_Click(object sender, EventArgs e)
    {
        ResetTimer();
    }

    private void txbNome_TextChanged(object sender, EventArgs e)
    {
        ResetTimer();
    }

```

```

}

private void lblNome_Click(object sender, EventArgs e)
{
    ResetTimer();
}

private void CadastroUsuario_Click(object sender, EventArgs e)
{
    ResetTimer();
}

#endregion

private void txbNome_Enter(object sender, EventArgs e)
{
    lblNome.Location = new Point(12, 95);
    txbNome.Location = new Point(12, 153);
    txbNome.Size = new Size(597, 54);

    lblMensagem.Location = new Point(550, 210);

    lblIdade.Location = new Point(741, 95);
    txbIdade.Location = new Point(741, 153);
    txbIdade.Size = new Size(597, 54);

    pcbKeyA.Visible = true;
    pcbKeyB.Visible = true;
    pcbKeyC.Visible = true;
    pcbKeyD.Visible = true;
    pcbKeyE.Visible = true;
    pcbKeyF.Visible = true;
    pcbKeyG.Visible = true;
    pcbKeyH.Visible = true;
    pcbKeyI.Visible = true;
    pcbKeyJ.Visible = true;
    pcbKeyK.Visible = true;
    pcbKeyL.Visible = true;
    pcbKeyM.Visible = true;
    pcbKeyN.Visible = true;
    pcbKeyO.Visible = true;
    pcbKeyP.Visible = true;
    pcbKeyQ.Visible = true;
    pcbKeyR.Visible = true;
    pcbKeyS.Visible = true;
    pcbKeyT.Visible = true;
    pcbKeyU.Visible = true;
    pcbKeyV.Visible = true;
    pcbKeyW.Visible = true;
    pcbKeyX.Visible = true;
    pcbKeyY.Visible = true;
    pcbKeyZ.Visible = true;
    pcbKeyÇ.Visible = true;
    pcbKeyShift.Visible = true;
    pcbKeySpace.Visible = true;
    pcbKeyBackspace.Visible = true;
    pcbKeyComma.Visible = true;
    pcbKeyDot.Visible = true;
}

private void txbNome_Leave(object sender, EventArgs e)
{

```

```

pcbKeyA.Visible = false;
pcbKeyB.Visible = false;
pcbKeyC.Visible = false;
pcbKeyD.Visible = false;
pcbKeyE.Visible = false;
pcbKeyF.Visible = false;
pcbKeyG.Visible = false;
pcbKeyH.Visible = false;
pcbKeyI.Visible = false;
pcbKeyJ.Visible = false;
pcbKeyK.Visible = false;
pcbKeyL.Visible = false;
pcbKeyM.Visible = false;
pcbKeyN.Visible = false;
pcbKeyO.Visible = false;
pcbKeyP.Visible = false;
pcbKeyQ.Visible = false;
pcbKeyR.Visible = false;
pcbKeyS.Visible = false;
pcbKeyT.Visible = false;
pcbKeyU.Visible = false;
pcbKeyV.Visible = false;
pcbKeyW.Visible = false;
pcbKeyX.Visible = false;
pcbKeyY.Visible = false;
pcbKeyZ.Visible = false;
pcbKeyÇ.Visible = false;
pcbKeyShift.Visible = false;
pcbKeySpace.Visible = false;
pcbKeyBackspace.Visible = false;
pcbKeyComma.Visible = false;
pcbKeyDot.Visible = false;
}

private void txbIdade_Enter(object sender, EventArgs e)
{
    lblNome.Location = new Point(12, 95);
    txbNome.Location = new Point(12, 153);
    txbNome.Size = new Size(597, 54);

    lblMensagem.Location = new Point(550, 210);

    lblIdade.Location = new Point(741, 95);
    txbIdade.Location = new Point(741, 153);
    txbIdade.Size = new Size(597, 54);

    pcbNumpad0.Visible = true;
    pcbNumpad1.Visible = true;
    pcbNumpad2.Visible = true;
    pcbNumpad3.Visible = true;
    pcbNumpad4.Visible = true;
    pcbNumpad5.Visible = true;
    pcbNumpad6.Visible = true;
    pcbNumpad7.Visible = true;
    pcbNumpad8.Visible = true;
    pcbNumpad9.Visible = true;
    pcbNumpadBackspace.Visible = true;
}

private void txbIdade_Leave(object sender, EventArgs e)

```

```

{
    pcbNumpad0.Visible = false;
    pcbNumpad1.Visible = false;
    pcbNumpad2.Visible = false;
    pcbNumpad3.Visible = false;
    pcbNumpad4.Visible = false;
    pcbNumpad5.Visible = false;
    pcbNumpad6.Visible = false;
    pcbNumpad7.Visible = false;
    pcbNumpad8.Visible = false;
    pcbNumpad9.Visible = false;
    pcbNumpadBackspace.Visible = false;
}

#region Teclados

#region Keyboard

public void AtualizarTeclas(int shiftOpcao)
{
    switch (shiftOpcao)
    {
        case 0:
            pcbKeyA.ImageLocation = "Imagens/Teclas/Minusculo/a.png";
            pcbKeyB.ImageLocation = "Imagens/Teclas/Minusculo/b.png";
            pcbKeyC.ImageLocation = "Imagens/Teclas/Minusculo/c.png";
            pcbKeyD.ImageLocation = "Imagens/Teclas/Minusculo/d.png";
            pcbKeyE.ImageLocation = "Imagens/Teclas/Minusculo/e.png";
            pcbKeyF.ImageLocation = "Imagens/Teclas/Minusculo/f.png";
            pcbKeyG.ImageLocation = "Imagens/Teclas/Minusculo/g.png";
            pcbKeyH.ImageLocation = "Imagens/Teclas/Minusculo/h.png";
            pcbKeyI.ImageLocation = "Imagens/Teclas/Minusculo/i.png";
            pcbKeyJ.ImageLocation = "Imagens/Teclas/Minusculo/j.png";
            pcbKeyK.ImageLocation = "Imagens/Teclas/Minusculo/k.png";
            pcbKeyL.ImageLocation = "Imagens/Teclas/Minusculo/l.png";
            pcbKeyM.ImageLocation = "Imagens/Teclas/Minusculo/m.png";
            pcbKeyN.ImageLocation = "Imagens/Teclas/Minusculo/n.png";
            pcbKeyO.ImageLocation = "Imagens/Teclas/Minusculo/o.png";
            pcbKeyP.ImageLocation = "Imagens/Teclas/Minusculo/p.png";
            pcbKeyQ.ImageLocation = "Imagens/Teclas/Minusculo/q.png";
            pcbKeyR.ImageLocation = "Imagens/Teclas/Minusculo/r.png";
            pcbKeyS.ImageLocation = "Imagens/Teclas/Minusculo/s.png";
            pcbKeyT.ImageLocation = "Imagens/Teclas/Minusculo/t.png";
            pcbKeyU.ImageLocation = "Imagens/Teclas/Minusculo/u.png";
            pcbKeyV.ImageLocation = "Imagens/Teclas/Minusculo/v.png";
            pcbKeyW.ImageLocation = "Imagens/Teclas/Minusculo/w.png";
            pcbKeyX.ImageLocation = "Imagens/Teclas/Minusculo/x.png";
            pcbKeyY.ImageLocation = "Imagens/Teclas/Minusculo/y.png";
            pcbKeyZ.ImageLocation = "Imagens/Teclas/Minusculo/z.png";
            pcbKeyÇ.ImageLocation = "Imagens/Teclas/Minusculo/ç.png";
            pcbKeyShift.ImageLocation =
"Imagens/Teclas/ShiftDesligado.png";
            break;

        case 1:
            pcbKeyA.ImageLocation = "Imagens/Teclas/Maiusculas/A.png";
            pcbKeyB.ImageLocation = "Imagens/Teclas/Maiusculas/B.png";
            pcbKeyC.ImageLocation = "Imagens/Teclas/Maiusculas/C.png";
            pcbKeyD.ImageLocation = "Imagens/Teclas/Maiusculas/D.png";
            pcbKeyE.ImageLocation = "Imagens/Teclas/Maiusculas/E.png";
            pcbKeyF.ImageLocation = "Imagens/Teclas/Maiusculas/F.png";

```

```

        pcbKeyG.ImageLocation = "Imagens/Tecclas/Maiusculas/G.png";
        pcbKeyH.ImageLocation = "Imagens/Tecclas/Maiusculas/H.png";
        pcbKeyI.ImageLocation = "Imagens/Tecclas/Maiusculas/I.png";
        pcbKeyJ.ImageLocation = "Imagens/Tecclas/Maiusculas/J.png";
        pcbKeyK.ImageLocation = "Imagens/Tecclas/Maiusculas/K.png";
        pcbKeyL.ImageLocation = "Imagens/Tecclas/Maiusculas/L.png";
        pcbKeyM.ImageLocation = "Imagens/Tecclas/Maiusculas/M.png";
        pcbKeyN.ImageLocation = "Imagens/Tecclas/Maiusculas/N.png";
        pcbKeyO.ImageLocation = "Imagens/Tecclas/Maiusculas/O.png";
        pcbKeyP.ImageLocation = "Imagens/Tecclas/Maiusculas/P.png";
        pcbKeyQ.ImageLocation = "Imagens/Tecclas/Maiusculas/Q.png";
        pcbKeyR.ImageLocation = "Imagens/Tecclas/Maiusculas/R.png";
        pcbKeyS.ImageLocation = "Imagens/Tecclas/Maiusculas/S.png";
        pcbKeyT.ImageLocation = "Imagens/Tecclas/Maiusculas/T.png";
        pcbKeyU.ImageLocation = "Imagens/Tecclas/Maiusculas/U.png";
        pcbKeyV.ImageLocation = "Imagens/Tecclas/Maiusculas/V.png";
        pcbKeyW.ImageLocation = "Imagens/Tecclas/Maiusculas/W.png";
        pcbKeyX.ImageLocation = "Imagens/Tecclas/Maiusculas/X.png";
        pcbKeyY.ImageLocation = "Imagens/Tecclas/Maiusculas/Y.png";
        pcbKeyZ.ImageLocation = "Imagens/Tecclas/Maiusculas/Z.png";
        pcbKeyÇ.ImageLocation = "Imagens/Tecclas/Maiusculas/Ç.png";
        pcbKeyShift.ImageLocation = "Imagens/Tecclas/ShiftAtivo.png";
        break;
    }
}

private void pcbKeyBackspace_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length > 0)
    {
        txbNome.Text = txbNome.Text.Substring(0, txbNome.Text.Length -
1);
    }
}

private void pcbKeyA_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(27);
                break;

            case 1:
                txbNome.Text += controle.Teclado(1);
                break;
        }
    }
}

private void pcbKeyB_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)

```

```

        {
            case 0:
                txbNome.Text += controle.Teclado(28);
                break;

            case 1:
                txbNome.Text += controle.Teclado(2);
                break;
        }
    }
}

private void pcbKeyC_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(29);
                break;

            case 1:
                txbNome.Text += controle.Teclado(3);
                break;
        }
    }
}

private void pcbKeyD_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(30);
                break;

            case 1:
                txbNome.Text += controle.Teclado(4);
                break;
        }
    }
}

private void pcbKeyE_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(31);
                break;

            case 1:

```

```

        txbNome.Text += controle.Teclado(5);
        break;
    }
}

private void pcbKeyF_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(32);
                break;

            case 1:
                txbNome.Text += controle.Teclado(6);
                break;
        }
    }
}

private void pcbKeyG_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(33);
                break;

            case 1:
                txbNome.Text += controle.Teclado(7);
                break;
        }
    }
}

private void pcbKeyH_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(34);
                break;

            case 1:
                txbNome.Text += controle.Teclado(8);
                break;
        }
    }
}

```



```

private void pcbKeyI_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(35);
                break;

            case 1:
                txbNome.Text += controle.Teclado(9);
                break;
        }
    }
}

private void pcbKeyJ_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(36);
                break;

            case 1:
                txbNome.Text += controle.Teclado(10);
                break;
        }
    }
}

private void pcbKeyK_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(37);
                break;

            case 1:
                txbNome.Text += controle.Teclado(11);
                break;
        }
    }
}

private void pcbKeyL_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
    }
}

```

```

        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(38);
                break;

            case 1:
                txbNome.Text += controle.Teclado(12);
                break;
        }
    }
}

private void pcbKeyM_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(39);
                break;

            case 1:
                txbNome.Text += controle.Teclado(13);
                break;
        }
    }
}

private void pcbKeyN_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(40);
                break;

            case 1:
                txbNome.Text += controle.Teclado(14);
                break;
        }
    }
}

private void pcbKeyO_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(41);
                break;

```

```

        case 1:
            txbNome.Text += controle.Teclado(15);
            break;
    }
}

private void pcbKeyP_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(42);
                break;

            case 1:
                txbNome.Text += controle.Teclado(16);
                break;
        }
    }
}

private void pcbKeyQ_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(43);
                break;

            case 1:
                txbNome.Text += controle.Teclado(17);
                break;
        }
    }
}

private void pcbKeyR_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(44);
                break;

            case 1:
                txbNome.Text += controle.Teclado(18);
                break;
        }
    }
}

```

```

private void pcbKeyS_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(45);
                break;

            case 1:
                txbNome.Text += controle.Teclado(19);
                break;
        }
    }
}

private void pcbKeyT_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(46);
                break;

            case 1:
                txbNome.Text += controle.Teclado(20);
                break;
        }
    }
}

private void pcbKeyU_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(47);
                break;

            case 1:
                txbNome.Text += controle.Teclado(21);
                break;
        }
    }
}

private void pcbKeyV_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {

```

```

        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(48);
                break;

            case 1:
                txbNome.Text += controle.Teclado(22);
                break;
        }
    }
}

private void pcbKeyW_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(49);
                break;

            case 1:
                txbNome.Text += controle.Teclado(23);
                break;
        }
    }
}

private void pcbKeyX_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(50);
                break;

            case 1:
                txbNome.Text += controle.Teclado(24);
                break;
        }
    }
}

private void pcbKeyY_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(51);
                break;
        }
    }
}

```

```

        case 1:
            txbNome.Text += controle.Teclado(25);
            break;
    }
}

private void pcbKeyZ_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(52);
                break;

            case 1:
                txbNome.Text += controle.Teclado(26);
                break;
        }
    }
}

private void pcbKeyÇ_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbNome.Text += controle.Teclado(57);
                break;

            case 1:
                txbNome.Text += controle.Teclado(56);
                break;
        }
    }
}

private void pcbKeyComma_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        txbNome.Text += controle.Teclado(53);
    }
}

private void pcbKeyDot_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        txbNome.Text += controle.Teclado(54);
    }
}

```

```

    }
}

private void pcbKeySpace_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbNome.Text.Length < txbNome.MaxLength)
    {
        Controle controle = new Controle();
        txbNome.Text += controle.Teclado(55);
    }
}

private void pcbKeyShift_Click(object sender, EventArgs e)
{
    ResetTimer();
    switch (shiftOpcao)
    {
        case 0:
            shiftOpcao = 1;
            break;

        case 1:
            shiftOpcao = 0;
            break;

        default:
            shiftOpcao = 1;
            break;
    }
    AtualizarTeclas(shiftOpcao);
}
#endregion

#region Numpad
private void pcbNumpad1_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbIdade.Text.Length < txbIdade.MaxLength)
    {
        Controle controle = new Controle();
        txbIdade.Text += controle.Teclado(58);
    }
}

private void pcbNumpad2_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbIdade.Text.Length < txbIdade.MaxLength)
    {
        Controle controle = new Controle();
        txbIdade.Text += controle.Teclado(59);
    }
}

private void pcbNumpad3_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbIdade.Text.Length < txbIdade.MaxLength)
    {
        Controle controle = new Controle();
        txbIdade.Text += controle.Teclado(60);
    }
}

```

```

}

private void pcbNumpad4_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbIdade.Text.Length < txbIdade.MaxLength)
    {
        Controle controle = new Controle();
        txbIdade.Text += controle.Teclado(61);
    }
}

private void pcbNumpad5_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbIdade.Text.Length < txbIdade.MaxLength)
    {
        Controle controle = new Controle();
        txbIdade.Text += controle.Teclado(62);
    }
}

private void pcbNumpad6_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbIdade.Text.Length < txbIdade.MaxLength)
    {
        Controle controle = new Controle();
        txbIdade.Text += controle.Teclado(63);
    }
}

private void pcbNumpad7_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbIdade.Text.Length < txbIdade.MaxLength)
    {
        Controle controle = new Controle();
        txbIdade.Text += controle.Teclado(64);
    }
}

private void pcbNumpad8_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbIdade.Text.Length < txbIdade.MaxLength)
    {
        Controle controle = new Controle();
        txbIdade.Text += controle.Teclado(65);
    }
}

private void pcbNumpad9_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbIdade.Text.Length < txbIdade.MaxLength)
    {
        Controle controle = new Controle();
        txbIdade.Text += controle.Teclado(66);
    }
}

private void pcbNumpad0_Click(object sender, EventArgs e)

```



```

    {
        ResetTimer();
        if (txbIdade.Text.Length < txbIdade.MaxLength)
        {
            Controle controle = new Controle();
            txbIdade.Text += controle.Teclado(67);
        }
    }

private void pcbNumpadBackspace_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbIdade.Text.Length > 0)
    {
        txbIdade.Text = txbIdade.Text.Substring(0, txbIdade.Text.Length -
1);
    }
}
#endregion

#endregion

    }
}

```

APÊNDICE H – AVALIACAOMUSEU

```

using SIstemaTotemHomemALua.Modelos;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace SIstemaTotemHomemALua
{
    public partial class AvaliacaoMuseu : Form
    {
        public AvaliacaoMuseu()
        {
            InitializeComponent();
            notaAvaliacao = 0;
            perguntaId = 1;
            shiftOpcao = 1;
            txbComentario.Text = "";
            AvancarPergunta();
            AtualizarTeclas(shiftOpcao);
        }

        public string mensagemForm1 = "";
        int perguntaId = 1;
        int notaQualidadeObras = 0;
        int notaVisitaMuseu = 0;
        int notaRecomendacao = 0;
        int notaAvaliacao = 0;
        int shiftOpcao = 1;

        #region Luas Cliques
        private void pcbLua1_Click(object sender, EventArgs e)
        {
            ResetTimer();
            AtualizarNotas(1);

            notaAvaliacao = 1;
        }

        private void pcbLua2_Click(object sender, EventArgs e)
        {
            ResetTimer();
            AtualizarNotas(2);

            notaAvaliacao = 2;
        }

        private void pcbLua3_Click(object sender, EventArgs e)
        {
            ResetTimer();
            AtualizarNotas(3);

            notaAvaliacao = 3;
        }

        private void pcbLua4_Click(object sender, EventArgs e)

```

```

    {
        ResetTimer();
        AtualizarNotas(4);

        notaAvaliacao = 4;
    }

private void pcbLua5_Click(object sender, EventArgs e)
{
    ResetTimer();
    AtualizarNotas(5);

    notaAvaliacao = 5;
}
#endregion

public void AtualizarNotas(int NotaImagem)
{
    switch (NotaImagem)
    {
        case 0:
            pcbLua1.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoDES.png";
            pcbLua2.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoDES.png";
            pcbLua3.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoDES.png";
            pcbLua4.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoDES.png";
            pcbLua5.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoDES.png";
            break;

        case 1:
            pcbLua1.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoATI.png";

            pcbLua2.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoDES.png";
            pcbLua3.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoDES.png";
            pcbLua4.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoDES.png";
            pcbLua5.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoDES.png";
            break;

        case 2:
            pcbLua1.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoATI.png";
            pcbLua2.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoATI.png";

            pcbLua3.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoDES.png";
            pcbLua4.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoDES.png";
            pcbLua5.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoDES.png";
            break;

        case 3:

```

```

        pcbLua1.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoATI.png";
        pcbLua2.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoATI.png";
        pcbLua3.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoATI.png";

        pcbLua4.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoDES.png";
        pcbLua5.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoDES.png";
        break;

        case 4:
            pcbLua1.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoATI.png";
            pcbLua2.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoATI.png";
            pcbLua3.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoATI.png";
            pcbLua4.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoATI.png";
            pcbLua5.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoDES.png";
            break;

        case 5:
            pcbLua1.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoATI.png";
            pcbLua2.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoATI.png";
            pcbLua3.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoATI.png";
            pcbLua4.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoATI.png";
            pcbLua5.ImageLocation =
"Imagens/Botões/BotaoAvaliacaoATI.png";
            break;

        default:
            NotaImagem = 0;
            AtualizarNotas(NotaImagem);
            break;
    }
}

private void AvancarPergunta()
{
    switch (perguntaId)
    {
        case 1:
            notaAvaliacao = 0;
            lblPergunta.Text = "Como você classificaria a qualidade
das\nobras de arte representadas no museu?";
            AtualizarNotas(0);

            txbComentario.Visible = false;
            pcbLua1.Visible = true;
            pcbLua2.Visible = true;
            pcbLua3.Visible = true;
            pcbLua4.Visible = true;
            pcbLua5.Visible = true;

```

```

lblMensagem.Visible = false;
lblMensagem.Text = "";

pcbKeyA.Visible = false;
pcbKeyB.Visible = false;
pcbKeyC.Visible = false;
pcbKeyD.Visible = false;
pcbKeyE.Visible = false;
pcbKeyF.Visible = false;
pcbKeyG.Visible = false;
pcbKeyH.Visible = false;
pcbKeyI.Visible = false;
pcbKeyJ.Visible = false;
pcbKeyK.Visible = false;
pcbKeyL.Visible = false;
pcbKeyM.Visible = false;
pcbKeyN.Visible = false;
pcbKeyO.Visible = false;
pcbKeyP.Visible = false;
pcbKeyQ.Visible = false;
pcbKeyR.Visible = false;
pcbKeyS.Visible = false;
pcbKeyT.Visible = false;
pcbKeyU.Visible = false;
pcbKeyV.Visible = false;
pcbKeyW.Visible = false;
pcbKeyX.Visible = false;
pcbKeyY.Visible = false;
pcbKeyZ.Visible = false;
pcbKeyÇ.Visible = false;
pcbKeyShift.Visible = false;
pcbKeySpace.Visible = false;
pcbKeyBackspace.Visible = false;
pcbKeyComma.Visible = false;
pcbKeyDot.Visible = false;
break;

case 2:
    notaAvaliacao = 0;
    lblPergunta.Text = "Como você avaliaria sua
experiência\ndurante a visita ao museu?";
    AtualizarNotas(0);

    txbComentario.Visible = false;
    pcbLua1.Visible = true;
    pcbLua2.Visible = true;
    pcbLua3.Visible = true;
    pcbLua4.Visible = true;
    pcbLua5.Visible = true;

    lblMensagem.Visible = false;
    lblMensagem.Text = "";

    pcbKeyA.Visible = false;
    pcbKeyB.Visible = false;
    pcbKeyC.Visible = false;
    pcbKeyD.Visible = false;
    pcbKeyE.Visible = false;
    pcbKeyF.Visible = false;
    pcbKeyG.Visible = false;
    pcbKeyH.Visible = false;
    pcbKeyI.Visible = false;

```

```

pcbKeyJ.Visible = false;
pcbKeyK.Visible = false;
pcbKeyL.Visible = false;
pcbKeyM.Visible = false;
pcbKeyN.Visible = false;
pcbKeyO.Visible = false;
pcbKeyP.Visible = false;
pcbKeyQ.Visible = false;
pcbKeyR.Visible = false;
pcbKeyS.Visible = false;
pcbKeyT.Visible = false;
pcbKeyU.Visible = false;
pcbKeyV.Visible = false;
pcbKeyW.Visible = false;
pcbKeyX.Visible = false;
pcbKeyY.Visible = false;
pcbKeyZ.Visible = false;
pcbKeyÇ.Visible = false;
pcbKeyShift.Visible = false;
pcbKeySpace.Visible = false;
pcbKeyBackspace.Visible = false;
pcbKeyComma.Visible = false;
pcbKeyDot.Visible = false;
break;

case 3:
    notaAvaliacao = 0;
    lblPergunta.Text = "Em uma escala de 1 a 5, o quanto
você\nrecomendaria esta visita a um\nparente ou amigo?";
    AtualizarNotas(0);

    txbComentario.Visible = false;
    pcbLua1.Visible = true;
    pcbLua2.Visible = true;
    pcbLua3.Visible = true;
    pcbLua4.Visible = true;
    pcbLua5.Visible = true;

    lblMensagem.Visible = false;
    lblMensagem.Text = "";

    pcbKeyA.Visible = false;
    pcbKeyB.Visible = false;
    pcbKeyC.Visible = false;
    pcbKeyD.Visible = false;
    pcbKeyE.Visible = false;
    pcbKeyF.Visible = false;
    pcbKeyG.Visible = false;
    pcbKeyH.Visible = false;
    pcbKeyI.Visible = false;
    pcbKeyJ.Visible = false;
    pcbKeyK.Visible = false;
    pcbKeyL.Visible = false;
    pcbKeyM.Visible = false;
    pcbKeyN.Visible = false;
    pcbKeyO.Visible = false;
    pcbKeyP.Visible = false;
    pcbKeyQ.Visible = false;
    pcbKeyR.Visible = false;
    pcbKeyS.Visible = false;
    pcbKeyT.Visible = false;
    pcbKeyU.Visible = false;
    pcbKeyV.Visible = false;

```

```

pcbKeyW.Visible = false;
pcbKeyX.Visible = false;
pcbKeyY.Visible = false;
pcbKeyZ.Visible = false;
pcbKeyÇ.Visible = false;
pcbKeyShift.Visible = false;
pcbKeySpace.Visible = false;
pcbKeyBackspace.Visible = false;
pcbKeyComma.Visible = false;
pcbKeyDot.Visible = false;
break;

case 4:
    notaAvaliacao = 1;
    lblPergunta.Text = "Sua opinião é importante para nós.\nDeixe
seu comentário! (Opcional)";
    AtualizarNotas(0);

    txbComentario.Visible = true;
    pcbLua1.Visible = false;
    pcbLua2.Visible = false;
    pcbLua3.Visible = false;
    pcbLua4.Visible = false;
    pcbLua5.Visible = false;

    lblMensagem.Visible = false;
    lblMensagem.Text = "";

    pcbKeyA.Visible = true;
    pcbKeyB.Visible = true;
    pcbKeyC.Visible = true;
    pcbKeyD.Visible = true;
    pcbKeyE.Visible = true;
    pcbKeyF.Visible = true;
    pcbKeyG.Visible = true;
    pcbKeyH.Visible = true;
    pcbKeyI.Visible = true;
    pcbKeyJ.Visible = true;
    pcbKeyK.Visible = true;
    pcbKeyL.Visible = true;
    pcbKeyM.Visible = true;
    pcbKeyN.Visible = true;
    pcbKeyO.Visible = true;
    pcbKeyP.Visible = true;
    pcbKeyQ.Visible = true;
    pcbKeyR.Visible = true;
    pcbKeyS.Visible = true;
    pcbKeyT.Visible = true;
    pcbKeyU.Visible = true;
    pcbKeyV.Visible = true;
    pcbKeyW.Visible = true;
    pcbKeyX.Visible = true;
    pcbKeyY.Visible = true;
    pcbKeyZ.Visible = true;
    pcbKeyÇ.Visible = true;
    pcbKeyShift.Visible = true;
    pcbKeySpace.Visible = true;
    pcbKeyBackspace.Visible = true;
    pcbKeyComma.Visible = true;
    pcbKeyDot.Visible = true;
    break;

case 5:

```

```

        tmrTempoAusencia.Enabled = false;
        Controle controle = new Controle();
        controle.ConcluirCadastro(notaQualidadeObras,
notaVisitaMuseu, notaRecomendacao, txbComentario.Text);
        if (controle.MensagemForm1.Equals("Armazenar"))
        {
            mensagemForm1 = controle.MensagemForm1;
        }
        this.Close();
        break;

        default:
            perguntaId = 1;
            AvancarPergunta();
            break;
    }
}

private void pbxBotaoContinuar_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (notaAvaliacao.Equals(0))
    {
        lblMensagem.Visible = true;
        lblMensagem.Text = "Nota não foi atribuida";
    }
    else
    {
        switch (perguntaId)
        {
            case 1:
                notaQualidadeObras = notaAvaliacao;
                break;

            case 2:
                notaVisitaMuseu = notaAvaliacao;
                break;

            case 3:
                notaRecomendacao = notaAvaliacao;
                break;
        }
        perguntaId++;
        AvancarPergunta();
    }
}

#region Reset Timer Ausencia
public void ResetTimer()
{
    tmrTempoAusencia.Enabled = false;
    tmrTempoAusencia.Enabled = true;
}

private void tmrTempoAusencia_Tick(object sender, EventArgs e)
{
    this.Close();
}

private void txbComentario_TextChanged(object sender, EventArgs e)
{
    ResetTimer();
}

```



```

}

private void AvaliacaoMuseu_Click(object sender, EventArgs e)
{
    ResetTimer();
}

private void lblSubtitulo_Click(object sender, EventArgs e)
{
    ResetTimer();
}

private void lblTitulo_Click(object sender, EventArgs e)
{
    ResetTimer();
}
#endregion

#region Teclado

public void AtualizarTeclas(int shiftOpcao)
{
    switch (shiftOpcao)
    {
        case 0:
            pcbKeyA.ImageLocation = "Imagens/Tecclas/Minusculo/a.png";
            pcbKeyB.ImageLocation = "Imagens/Tecclas/Minusculo/b.png";
            pcbKeyC.ImageLocation = "Imagens/Tecclas/Minusculo/c.png";
            pcbKeyD.ImageLocation = "Imagens/Tecclas/Minusculo/d.png";
            pcbKeyE.ImageLocation = "Imagens/Tecclas/Minusculo/e.png";
            pcbKeyF.ImageLocation = "Imagens/Tecclas/Minusculo/f.png";
            pcbKeyG.ImageLocation = "Imagens/Tecclas/Minusculo/g.png";
            pcbKeyH.ImageLocation = "Imagens/Tecclas/Minusculo/h.png";
            pcbKeyI.ImageLocation = "Imagens/Tecclas/Minusculo/i.png";
            pcbKeyJ.ImageLocation = "Imagens/Tecclas/Minusculo/j.png";
            pcbKeyK.ImageLocation = "Imagens/Tecclas/Minusculo/k.png";
            pcbKeyL.ImageLocation = "Imagens/Tecclas/Minusculo/l.png";
            pcbKeyM.ImageLocation = "Imagens/Tecclas/Minusculo/m.png";
            pcbKeyN.ImageLocation = "Imagens/Tecclas/Minusculo/n.png";
            pcbKeyO.ImageLocation = "Imagens/Tecclas/Minusculo/o.png";
            pcbKeyP.ImageLocation = "Imagens/Tecclas/Minusculo/p.png";
            pcbKeyQ.ImageLocation = "Imagens/Tecclas/Minusculo/q.png";
            pcbKeyR.ImageLocation = "Imagens/Tecclas/Minusculo/r.png";
            pcbKeyS.ImageLocation = "Imagens/Tecclas/Minusculo/s.png";
            pcbKeyT.ImageLocation = "Imagens/Tecclas/Minusculo/t.png";
            pcbKeyU.ImageLocation = "Imagens/Tecclas/Minusculo/u.png";
            pcbKeyV.ImageLocation = "Imagens/Tecclas/Minusculo/v.png";
            pcbKeyW.ImageLocation = "Imagens/Tecclas/Minusculo/w.png";
            pcbKeyX.ImageLocation = "Imagens/Tecclas/Minusculo/x.png";
            pcbKeyY.ImageLocation = "Imagens/Tecclas/Minusculo/y.png";
            pcbKeyZ.ImageLocation = "Imagens/Tecclas/Minusculo/z.png";
            pcbKeyÇ.ImageLocation = "Imagens/Tecclas/Minusculo/ç.png";
            pcbKeyShift.ImageLocation =
"Imagens/Tecclas/ShiftDesligado.png";
            break;

        case 1:
            pcbKeyA.ImageLocation = "Imagens/Tecclas/Maiusculas/A.png";
            pcbKeyB.ImageLocation = "Imagens/Tecclas/Maiusculas/B.png";
            pcbKeyC.ImageLocation = "Imagens/Tecclas/Maiusculas/C.png";

```

```

        pcbKeyD.ImageLocation = "Imagens/Tecclas/Maiusculas/D.png";
        pcbKeyE.ImageLocation = "Imagens/Tecclas/Maiusculas/E.png";
        pcbKeyF.ImageLocation = "Imagens/Tecclas/Maiusculas/F.png";
        pcbKeyG.ImageLocation = "Imagens/Tecclas/Maiusculas/G.png";
        pcbKeyH.ImageLocation = "Imagens/Tecclas/Maiusculas/H.png";
        pcbKeyI.ImageLocation = "Imagens/Tecclas/Maiusculas/I.png";
        pcbKeyJ.ImageLocation = "Imagens/Tecclas/Maiusculas/J.png";
        pcbKeyK.ImageLocation = "Imagens/Tecclas/Maiusculas/K.png";
        pcbKeyL.ImageLocation = "Imagens/Tecclas/Maiusculas/L.png";
        pcbKeyM.ImageLocation = "Imagens/Tecclas/Maiusculas/M.png";
        pcbKeyN.ImageLocation = "Imagens/Tecclas/Maiusculas/N.png";
        pcbKeyO.ImageLocation = "Imagens/Tecclas/Maiusculas/O.png";
        pcbKeyP.ImageLocation = "Imagens/Tecclas/Maiusculas/P.png";
        pcbKeyQ.ImageLocation = "Imagens/Tecclas/Maiusculas/Q.png";
        pcbKeyR.ImageLocation = "Imagens/Tecclas/Maiusculas/R.png";
        pcbKeyS.ImageLocation = "Imagens/Tecclas/Maiusculas/S.png";
        pcbKeyT.ImageLocation = "Imagens/Tecclas/Maiusculas/T.png";
        pcbKeyU.ImageLocation = "Imagens/Tecclas/Maiusculas/U.png";
        pcbKeyV.ImageLocation = "Imagens/Tecclas/Maiusculas/V.png";
        pcbKeyW.ImageLocation = "Imagens/Tecclas/Maiusculas/W.png";
        pcbKeyX.ImageLocation = "Imagens/Tecclas/Maiusculas/X.png";
        pcbKeyY.ImageLocation = "Imagens/Tecclas/Maiusculas/Y.png";
        pcbKeyZ.ImageLocation = "Imagens/Tecclas/Maiusculas/Z.png";
        pcbKeyÇ.ImageLocation = "Imagens/Tecclas/Maiusculas/Ç.png";
        pcbKeyShift.ImageLocation = "Imagens/Tecclas/ShiftAtivo.png";
        break;
    }
}

private void pcbKeyBackspace_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length > 0)
    {
        txbComentario.Text = txbComentario.Text.Substring(0,
txbComentario.Text.Length - 1);
    }
}

private void pcbKeyA_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(27);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(1);
                break;
        }
    }
}

private void pcbKeyB_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)

```

```

    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(28);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(2);
                break;
        }
    }
}

private void pcbKeyC_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(29);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(3);
                break;
        }
    }
}

private void pcbKeyD_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(30);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(4);
                break;
        }
    }
}

private void pcbKeyE_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(31);

```

```

        break;

        case 1:
            txbComentario.Text += controle.Teclado(5);
            break;
    }
}

private void pcbKeyF_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(32);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(6);
                break;
        }
    }
}

private void pcbKeyG_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(33);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(7);
                break;
        }
    }
}

private void pcbKeyH_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(34);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(8);
                break;
        }
    }
}

```

```

    }
}

private void pcbKeyI_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(35);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(9);
                break;
        }
    }
}

private void pcbKeyJ_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(36);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(10);
                break;
        }
    }
}

private void pcbKeyK_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(37);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(11);
                break;
        }
    }
}

private void pcbKeyL_Click(object sender, EventArgs e)
{
    ResetTimer();

```

```

        if (txbComentario.Text.Length < txbComentario.MaxLength)
        {
            Controle controle = new Controle();
            switch (shiftOpcao)
            {
                case 0:
                    txbComentario.Text += controle.Teclado(38);
                    break;

                case 1:
                    txbComentario.Text += controle.Teclado(12);
                    break;
            }
        }
    }

    private void pcbKeyM_Click(object sender, EventArgs e)
    {
        ResetTimer();
        if (txbComentario.Text.Length < txbComentario.MaxLength)
        {
            Controle controle = new Controle();
            switch (shiftOpcao)
            {
                case 0:
                    txbComentario.Text += controle.Teclado(39);
                    break;

                case 1:
                    txbComentario.Text += controle.Teclado(13);
                    break;
            }
        }
    }

    private void pcbKeyN_Click(object sender, EventArgs e)
    {
        ResetTimer();
        if (txbComentario.Text.Length < txbComentario.MaxLength)
        {
            Controle controle = new Controle();
            switch (shiftOpcao)
            {
                case 0:
                    txbComentario.Text += controle.Teclado(40);
                    break;

                case 1:
                    txbComentario.Text += controle.Teclado(14);
                    break;
            }
        }
    }

    private void pcbKeyO_Click(object sender, EventArgs e)
    {
        ResetTimer();
        if (txbComentario.Text.Length < txbComentario.MaxLength)
        {
            Controle controle = new Controle();
            switch (shiftOpcao)
            {
                case 0:

```

```

        txbComentario.Text += controle.Teclado(41);
        break;

        case 1:
            txbComentario.Text += controle.Teclado(15);
            break;
    }
}

private void pcbKeyP_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(42);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(16);
                break;
        }
    }
}

private void pcbKeyQ_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(43);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(17);
                break;
        }
    }
}

private void pcbKeyR_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(44);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(18);
                break;
        }
    }
}

```

```

    }
}

private void pcbKeyS_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(45);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(19);
                break;
        }
    }
}

private void pcbKeyT_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(46);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(20);
                break;
        }
    }
}

private void pcbKeyU_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(47);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(21);
                break;
        }
    }
}

private void pcbKeyV_Click(object sender, EventArgs e)
{

```



```

ResetTimer();
if (txbComentario.Text.Length < txbComentario.MaxLength)
{
    Controle controle = new Controle();
    switch (shiftOpcao)
    {
        case 0:
            txbComentario.Text += controle.Teclado(48);
            break;

        case 1:
            txbComentario.Text += controle.Teclado(22);
            break;
    }
}

private void pcbKeyW_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(49);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(23);
                break;
        }
    }
}

private void pcbKeyX_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(50);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(24);
                break;
        }
    }
}

private void pcbKeyY_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {

```

```

        case 0:
            txbComentario.Text += controle.Teclado(51);
            break;

        case 1:
            txbComentario.Text += controle.Teclado(25);
            break;
    }
}

private void pcbKeyZ_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(52);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(26);
                break;
        }
    }
}

private void pcbKeyÇ_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        switch (shiftOpcao)
        {
            case 0:
                txbComentario.Text += controle.Teclado(57);
                break;

            case 1:
                txbComentario.Text += controle.Teclado(56);
                break;
        }
    }
}

private void pcbKeyComma_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        txbComentario.Text += controle.Teclado(53);
    }
}

private void pcbKeyDot_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)

```

```

        {
            Controle controle = new Controle();
            txbComentario.Text += controle.Teclado(54);
        }
    }

private void pcbKeySpace_Click(object sender, EventArgs e)
{
    ResetTimer();
    if (txbComentario.Text.Length < txbComentario.MaxLength)
    {
        Controle controle = new Controle();
        txbComentario.Text += controle.Teclado(55);
    }
}

private void pcbKeyShift_Click(object sender, EventArgs e)
{
    ResetTimer();
    switch (shiftOpcao)
    {
        case 0:
            shiftOpcao = 1;
            break;

        case 1:
            shiftOpcao = 0;
            break;

        default:
            shiftOpcao = 1;
            break;
    }
    AtualizarTeclas(shiftOpcao);
}
#endregion
    }
}

```

APÊNDICE I – ESTATÍSTICAS

```

using SistemaTotemHomemALua.Modelos;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaTotemHomemALua
{
    public partial class Estatisticas : Form
    {
        public Estatisticas()
        {
            InitializeComponent();
            AtualizarEstatisticas();
            pagina = 1;
            ProximaPagina();
        }

        public void AtualizarEstatisticas()
        {
            //Jogou as respostas do usuario no Forms
            lblRespostaUsuario1.Text = Estatico.TEMPRESPOSTAQUEST1;
            lblRespostaUsuario2.Text = Estatico.TEMPRESPOSTAQUEST2;
            lblRespostaUsuario3.Text = Estatico.TEMPRESPOSTAQUEST3;
            lblRespostaUsuario4.Text = Estatico.TEMPRESPOSTAQUEST4;
            lblRespostaUsuario5.Text = Estatico.TEMPRESPOSTAQUEST5;

            lblTotalAcertos.Text = "Total De Acertos: " + Estatico.TOTALACERTOS;
            lblTotalErros.Text = "Total De Erros: " + Estatico.TOTALERROS;

            VerificarRespostas();

            AtualizarEstatisticaGlobal();
        }

        public string pergunta1 = "Acertou";
        public string pergunta2 = "Acertou";
        public string pergunta3 = "Acertou";
        public string pergunta4 = "Acertou";
        public string pergunta5 = "Acertou";
        public int pagina = 1;

        public void VerificarRespostas()
        {
            if (lblRespostaUsuario1.Text != lblRespostaCerta1.Text)
            {
                lblRespostaUsuario1.ForeColor = Color.IndianRed;
                lblRespostaUsuario1.Font = new Font(lblRespostaUsuario1.Font,
FontStyle.Strikeout);
                lblRespostaCerta1.Visible = true;
                pergunta1 = "Errou";
            }

            if (lblRespostaUsuario2.Text != lblRespostaCerta2.Text)
            {
                lblRespostaUsuario2.ForeColor = Color.IndianRed;
            }
        }
    }
}

```

```

        lblRespostaUsuario2.Font = new Font(lblRespostaUsuario2.Font,
FontStyle.Strikeout);
        lblRespostaCerta2.Visible = true;
        pergunta2 = "Errou";
    }

    if (lblRespostaUsuario3.Text != lblRespostaCerta3.Text)
    {
        lblRespostaUsuario3.ForeColor = Color.IndianRed;
        lblRespostaUsuario3.Font = new Font(lblRespostaUsuario3.Font,
FontStyle.Strikeout);
        lblRespostaCerta3.Visible = true;
        pergunta3 = "Errou";
    }

    if (lblRespostaUsuario4.Text != lblRespostaCerta4.Text)
    {
        lblRespostaUsuario4.ForeColor = Color.IndianRed;
        lblRespostaUsuario4.Font = new Font(lblRespostaUsuario4.Font,
FontStyle.Strikeout);
        lblRespostaCerta4.Visible = true;
        pergunta4 = "Errou";
    }

    if (lblRespostaUsuario5.Text != lblRespostaCerta5.Text)
    {
        lblRespostaUsuario5.ForeColor = Color.IndianRed;
        lblRespostaUsuario5.Font = new Font(lblRespostaUsuario5.Font,
FontStyle.Strikeout);
        lblRespostaCerta5.Visible = true;
        pergunta5 = "Errou";
    }
}

public void AtualizarEstatisticaGlobal()
{
    Random random = new Random();

    switch (pergunta1)
    {
        case "Acertou":
            int i = random.Next(1, 4);

            switch (i)
            {
                case 1:
                    lblTotalPessoasQuestao1.Text = Estatico.ACERTOSQUEST1
+ " Pessoa(s) acertaram esta questão, você esta é uma delas";
                    break;

                case 2:
                    lblTotalPessoasQuestao1.Text = "Você esta entre " +
Estatico.ACERTOSQUEST1 + " das pessoa(s) que acertaram essa pergunta";
                    break;

                case 3:
                    double porcentagem =
Math.Round((double)Estatico.ACERTOSQUEST1 / Estatico.TOTALRESPOSTASQUEST1 * 100,
1);

```

```

        lblTotalPessoasQuestao1.Text = "Você esta entre os "
+ porcentagem + "% que acertaram esta pergunta";
        break;

        default:
            MessageBox.Show("Sem frase\nNumeração: " + i);
            break;
    }

    break;

    case "Errou":

        int ii = random.Next(1, 4);

        switch (ii)
        {
            case 1:
                lblTotalPessoasQuestao1.Text = Estatico.ERROSQUEST1 +
" Pessoa(s) erraram esta questão";
                break;

            case 2:
                lblTotalPessoasQuestao1.Text = "Não se preocupe, " +
Estatico.ERROSQUEST1 + " pessoa(s) tambem erraram";
                break;

            case 3:
                double porcentagem =
Math.Round((double)Estatico.ERROSQUEST1 / Estatico.TOTALRESPOSTASQUEST1 * 100,
1);
                lblTotalPessoasQuestao1.Text = porcentagem + "% das
pessoas que passaram por esta pergunta, erraram";
                break;

            default:
                MessageBox.Show("Sem frase\nNumeração: " + ii);
                break;
        }

        break;
    }

    switch (pergunta2)
    {
        case "Acertou":

            int i = random.Next(1, 4);

            switch (i)
            {
                case 1:
                    lblTotalPessoasQuestao2.Text = Estatico.ACERTOSQUEST2
+ " Pessoa(s) acertaram esta questão, você esta é uma delas";
                    break;

                case 2:
                    lblTotalPessoasQuestao2.Text = "Você esta entre " +
Estatico.ACERTOSQUEST2 + " das pessoa(s) que acertaram essa pergunta";
                    break;

                case 3:

```

```

        double porcentagem =
Math.Round((double)Estatico.ACERTOSQUEST2 / Estatico.TOTALRESPOSTASQUEST2 * 100,
1);
        lblTotalPessoasQuestao2.Text = "Você esta entre os "
+ porcentagem + "% que acertaram esta pergunta";
        break;

        default:
            MessageBox.Show("Sem frase\nNumeração: " + i);
            break;
    }

    break;

    case "Errou":

        int ii = random.Next(1, 4);

        switch (ii)
        {
            case 1:
                lblTotalPessoasQuestao2.Text = Estatico.ERROSQUEST2 +
" Pessoa(s) erraram esta questão";
                break;

            case 2:
                lblTotalPessoasQuestao2.Text = "Não se preocupe, " +
Estatico.ERROSQUEST2 + " pessoa(s) tambem erraram";
                break;

            case 3:
                double porcentagem =
Math.Round((double)Estatico.ERROSQUEST2 / Estatico.TOTALRESPOSTASQUEST2 * 100,
1);
                lblTotalPessoasQuestao2.Text = porcentagem + "% das
pessoas que passaram por esta pergunta, erraram";
                break;

            default:
                MessageBox.Show("Sem frase\nNumeração: " + ii);
                break;
        }

        break;
    }

    switch (pergunta3)
    {
        case "Acertou":

            int i = random.Next(1, 4);

            switch (i)
            {
                case 1:
                    lblTotalPessoasQuestao3.Text = Estatico.ACERTOSQUEST3
+ " Pessoa(s) acertaram esta questão, você esta é uma delas";
                    break;

                case 2:
                    lblTotalPessoasQuestao3.Text = "Você esta entre " +
Estatico.ACERTOSQUEST3 + " das pessoa(s) que acertaram essa pergunta";

```

```

        break;

        case 3:
            double porcentagem =
Math.Round((double)Estatico.ACERTOSQUEST3 / Estatico.TOTALRESPOSTASQUEST3 * 100,
1);
            lblTotalPessoasQuestao3.Text = "Você esta entre os "
+ porcentagem + "% que acertaram esta pergunta";
            break;

        default:
            MessageBox.Show("Sem frase\nNumeração: " + i);
            break;
    }

    break;

    case "Errou":

        int ii = random.Next(1, 4);

        switch (ii)
        {
            case 1:
                lblTotalPessoasQuestao3.Text = Estatico.ERROSQUEST3 +
" Pessoa(s) erraram esta questão";
                break;

            case 2:
                lblTotalPessoasQuestao3.Text = "Não se preocupe, " +
Estatico.ERROSQUEST3 + " pessoa(s) tambem erraram";
                break;

            case 3:
                double porcentagem =
Math.Round((double)Estatico.ERROSQUEST3 / Estatico.TOTALRESPOSTASQUEST3 * 100,
1);
                lblTotalPessoasQuestao3.Text = porcentagem + "% das
pessoas que passaram por esta pergunta, erraram";
                break;

            default:
                MessageBox.Show("Sem frase\nNumeração: " + ii);
                break;
        }

        break;

    }

    switch (pergunta4)
    {
        case "Acertou":

            int i = random.Next(1, 4);

            switch (i)
            {
                case 1:
                    lblTotalPessoasQuestao4.Text = Estatico.ACERTOSQUEST4
+ " Pessoa(s) acertaram esta questão, você esta é uma delas";
                    break;

```



```

        case 2:
            lblTotalPessoasQuestao4.Text = "Você esta entre " +
Estatico.ACERTOSQUEST4 + " das pessoa(s) que acertaram essa pergunta";
            break;

        case 3:
            double porcentagem =
Math.Round((double)Estatico.ACERTOSQUEST4 / Estatico.TOTALRESPOSTASQUEST4 * 100,
1);
            lblTotalPessoasQuestao4.Text = "Você esta entre os "
+ porcentagem + "% que acertaram esta pergunta";
            break;

        default:
            MessageBox.Show("Sem frase\nNumeração: " + i);
            break;
    }

    break;

case "Errou":

    int ii = random.Next(1, 4);

    switch (ii)
    {
        case 1:
            lblTotalPessoasQuestao4.Text = Estatico.ERROSQUEST4 +
" Pessoa(s) erraram esta questão";
            break;

        case 2:
            lblTotalPessoasQuestao4.Text = "Não se preocupe, " +
Estatico.ERROSQUEST4 + " pessoa(s) tambem erraram";
            break;

        case 3:
            double porcentagem =
Math.Round((double)Estatico.ERROSQUEST4 / Estatico.TOTALRESPOSTASQUEST4 * 100,
1);
            lblTotalPessoasQuestao4.Text = porcentagem + "% das
pessoas que passaram por esta pergunta, erraram";
            break;

        default:
            MessageBox.Show("Sem frase\nNumeração: " + ii);
            break;
    }

    break;

}

switch (pergunta5)
{
    case "Acertou":

        int i = random.Next(1, 4);

        switch (i)
        {
            case 1:

```

```

        lblTotalPessoasQuestao5.Text = Estatico.ACERTOSQUEST5
+ " Pessoa(s) acertaram esta questão, você esta é uma delas";
        break;

        case 2:
            lblTotalPessoasQuestao5.Text = "Você esta entre " +
Estatico.ACERTOSQUEST5 + " das pessoa(s) que acertaram essa pergunta";
            break;

        case 3:
            double porcentagem =
Math.Round((double)Estatico.ACERTOSQUEST5 / Estatico.TOTALRESPOSTASQUEST5 * 100,
1);
            lblTotalPessoasQuestao5.Text = "Você esta entre os "
+ porcentagem + "% que acertaram esta pergunta";
            break;

        default:
            MessageBox.Show("Sem frase\nNumeração: " + i);
            break;
    }

    break;

case "Errou":

    int ii = random.Next(1, 4);

    switch (ii)
    {
        case 1:
            lblTotalPessoasQuestao5.Text = Estatico.ERROSQUEST5 +
" Pessoa(s) erraram esta questão";
            break;

        case 2:
            lblTotalPessoasQuestao5.Text = "Não se preocupe, " +
Estatico.ERROSQUEST5 + " pessoa(s) tambem erraram";
            break;

        case 3:
            double porcentagem =
Math.Round((double)Estatico.ERROSQUEST5 / Estatico.TOTALRESPOSTASQUEST5 * 100,
1);
            lblTotalPessoasQuestao5.Text = porcentagem + "% das
pessoas que passaram por esta pergunta, erraram";
            break;

        default:
            MessageBox.Show("Sem frase\nNumeração: " + ii);
            break;
    }

    break;

}

}

public void ProximaPagina()
{
    switch (pagina)
    {

```

case 1:

```

lblSuasRespostas.Visible = true;

lblPerguntaNumeracao1.Visible = true;
lblPerguntaNumeracao2.Visible = true;
lblPerguntaNumeracao3.Visible = true;
lblPerguntaNumeracao4.Visible = true;
lblPerguntaNumeracao5.Visible = true;

lblRespostaUsuario1.Visible = true;
lblRespostaUsuario2.Visible = true;
lblRespostaUsuario3.Visible = true;
lblRespostaUsuario4.Visible = true;
lblRespostaUsuario5.Visible = true;

lblEstatisticas.Visible = false;

lblTotalPessoasQuestao1.Visible = false;
lblTotalPessoasQuestao2.Visible = false;
lblTotalPessoasQuestao3.Visible = false;
lblTotalPessoasQuestao4.Visible = false;
lblTotalPessoasQuestao5.Visible = false;

lblTotalAcertos.Visible = false;
lblTotalErros.Visible = false;

break;

```

case 2:

```

lblSuasRespostas.Visible = false;

lblPerguntaNumeracao1.Visible = true;
lblPerguntaNumeracao2.Visible = true;
lblPerguntaNumeracao3.Visible = true;
lblPerguntaNumeracao4.Visible = true;
lblPerguntaNumeracao5.Visible = true;

lblPerguntaNumeracao1.Location = new Point(12, 84);
lblPerguntaNumeracao2.Location = new Point(12, 156);
lblPerguntaNumeracao3.Location = new Point(12, 229);
lblPerguntaNumeracao4.Location = new Point(12, 301);
lblPerguntaNumeracao5.Location = new Point(12, 377);

lblRespostaUsuario1.Visible = false;
lblRespostaUsuario2.Visible = false;
lblRespostaUsuario3.Visible = false;
lblRespostaUsuario4.Visible = false;
lblRespostaUsuario5.Visible = false;

lblRespostaCerta1.Visible = false;
lblRespostaCerta2.Visible = false;
lblRespostaCerta3.Visible = false;
lblRespostaCerta4.Visible = false;
lblRespostaCerta5.Visible = false;

```

```

        lblEstatisticas.Visible = true;
        lblEstatisticas.Location = new Point(12, 9);

        lblTotalPessoasQuestao1.Visible = true;
        lblTotalPessoasQuestao2.Visible = true;
        lblTotalPessoasQuestao3.Visible = true;
        lblTotalPessoasQuestao4.Visible = true;
        lblTotalPessoasQuestao5.Visible = true;

        lblTotalPessoasQuestao1.Location = new Point(91, 84 + 15);
        lblTotalPessoasQuestao2.Location = new Point(91, 156 + 15);
        lblTotalPessoasQuestao3.Location = new Point(91, 229 + 15);
        lblTotalPessoasQuestao4.Location = new Point(91, 301 + 15);
        lblTotalPessoasQuestao5.Location = new Point(91, 377 + 15);

        lblTotalAcertos.Visible = true;
        lblTotalErros.Visible = true;

        lblTotalAcertos.Location = new Point(12, 535);
        lblTotalErros.Location = new Point(12, 625);

        break;

        default:
            this.Close();
            break;
    }
}

private void pbxBotaoContinuar_Click(object sender, EventArgs e)
{
    ResetTimer();
    pagina++;
    ProximaPagina();
}

public void ResetTimer()
{
    tmrTempoAusencia.Enabled = false;
    tmrTempoAusencia.Enabled = true;
}

private void tmrTempoAusencia_Tick(object sender, EventArgs e)
{
    pagina++;
    ProximaPagina();
}

private void Estatisticas_Click(object sender, EventArgs e)
{
    ResetTimer();
}
}
}

```

APÊNDICE J – TABELADELIDERANCA

```

using SistemaTotemHomemALua.Modelos;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaTotemHomemALua
{
    public partial class TabelaDeLideranca : Form
    {
        List<Usuarios> usuarios = new List<Usuarios>();

        public TabelaDeLideranca(string caminhoArquivo)
        {
            InitializeComponent();

            JsonControle jsonControle = new JsonControle();
            usuarios = jsonControle.DeserializarLista(caminhoArquivo);
            usuarios = usuarios.OrderByDescending(u => u.QuestionarioNota).To-
List();

            int posicao = 1;
            foreach (var usuario in usuarios)
            {
                usuario.Posicao = posicao;
                posicao++;
            }

            dgvTabelaLideranca.DataSource = null;
            dgvTabelaLideranca.DataSource = usuarios;

            dgvTabelaLideranca.Columns["QuestionarioNota"].HeaderText = "Nota";

            dgvTabelaLideranca.Columns["Idade"].Visible = false;
            dgvTabelaLideranca.Columns["Comentario"].Visible = false;

        }

        private void pbxBotaoContinuar_Click(object sender, EventArgs e)
        {
            tmrTempoAusencia.Enabled = false;
            Agradecimento agr = new Agradecimento();
            agr.ShowDialog();
            this.Close();
        }

        #region Reset Timer Ausencia
        private void tmrTempoAusencia_Tick(object sender, EventArgs e)
        {
            Agradecimento agr = new Agradecimento();
            agr.ShowDialog();
        }
    }
}

```

```
        this.Close();
    }

    public void ResetTimer()
    {
        tmrTempoAusencia.Enabled = false;
        tmrTempoAusencia.Enabled = true;
    }

    private void TabelaDeLideranca_Click(object sender, EventArgs e)
    {
        ResetTimer();
    }

    private void lblRelatorio_Click(object sender, EventArgs e)
    {
        ResetTimer();
    }

    private void dgvRelatorio_Click(object sender, EventArgs e)
    {
        ResetTimer();
    }
    #endregion
}
}
```

APÊNDICE K – AGRADECIMENTO

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaTotemHomemALua
{
    public partial class Agradecimento : Form
    {
        public Agradecimento()
        {
            InitializeComponent();
        }

        private void Agradecimento_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void tmrFecharJanela_Tick(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}

```

APÊNDICE L – RELATORIO

```

using SistemaTotemHomemALua.Modelos;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Common;
using System.Diagnostics;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaTotemHomemALua
{
    public partial class Relatorio : Form
    {
        List<Usuarios> usuarios = new List<Usuarios>();

        public Relatorio(string caminhoArquivo)
        {
            InitializeComponent();
            if (File.Exists(caminhoArquivo))
            {
                JsonControle jsonControle = new JsonControle();
                usuarios = jsonControle.DeserializarLista(caminhoArquivo);
                dgvRelatorio.DataSource = usuarios;

                dgvRelatorio.Columns["Posicao"].Visible = false;
                dgvRelatorio.Columns["QuestionarioNota"].Visible = false;
            }
            else
            {
                lblMensagem.Visible = true;
                lblMensagem.Text = "Arquivo não encontrado";
            }
            AtualizarInformacoes();
        }

        public string mensagemForm1 = "";
        private string caminhoDiretorio = @"C:\LIT\BancoDeDados";

        private void pbxBotaoMenu_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void pcbMostrarBancoDeDados_Click(object sender, EventArgs e)
        {
            if (Directory.Exists(caminhoDiretorio))
            {
                Process.Start("explorer.exe", caminhoDiretorio);
            }
            else
            {
                lblMensagem.Visible = true;
                lblMensagem.Text = "Diretorio não existe";
            }
        }
    }
}

```



```

    }

    private void pcbFecharSistema_Click(object sender, EventArgs e)
    {
        mensagemForm1 = "FinalizarSistema";
        this.Close();
    }

    private void pcbRelatorioPerguntas_Click(object sender, EventArgs e)
    {
        RelatorioPerguntas relPerguntas = new RelatorioPerguntas();
        relPerguntas.ShowDialog();
    }

    public void AtualizarInformacoes()
    {
        if (!Estatico.SOMATOTALQUALIDADEOBRAS.Equals(0) && !Estatico.NUMEROAVALIACOESQUALIDADEOBRAS.Equals(0))
        {
            lblQualidadeObrasMedia.Text = Math.Round(((double)Estatico.SOMATOTALQUALIDADEOBRAS / Estatico.NUMEROAVALIACOESQUALIDADEOBRAS), 1).ToString();
        }
        else
        {
            lblQualidadeObrasMedia.Text = "0.0";
        }

        if (!Estatico.SOMATOTALVISITAMUSEU.Equals(0) && !Estatico.NUMEROAVALIACOESVISITAMUSEU.Equals(0))
        {
            lblExperienciaVisitaMedia.Text = Math.Round(((double)Estatico.SOMATOTALVISITAMUSEU / Estatico.NUMEROAVALIACOESVISITAMUSEU), 1).ToString();
        }
        else
        {
            lblExperienciaVisitaMedia.Text = "0.0";
        }

        if (!Estatico.SOMATOTALRECOMENDACAO.Equals(0) && !Estatico.NUMEROAVALIACOESRECOMENDACAO.Equals(0))
        {
            lblRecomendacaoMedia.Text = Math.Round(((double)Estatico.SOMATOTALRECOMENDACAO / Estatico.NUMEROAVALIACOESRECOMENDACAO), 1).ToString();
        }
        else
        {
            lblRecomendacaoMedia.Text = "0.0";
        }
    }
}
}

```

APÊNDICE M – RELATORIO PERGUNTAS

```

using SistemaTotemHomemALua.Modelos;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaTotemHomemALua
{
    public partial class RelatorioPerguntas : Form
    {
        public RelatorioPerguntas()
        {
            InitializeComponent();
            AtualizarInformacoes();
        }

        public void AtualizarInformacoes()
        {
            lblTotalRespostas1.Text += Estatico.TOTALRESPOSTASQUEST1;
            lblTotalAcertos1.Text += Estatico.ACERTOSQUEST1;
            lblTotalErros1.Text += Estatico.ERROSQUEST1;
            if (Estatico.ACERTOSQUEST1.Equals(0) && Estatico.ERROS-
QUEST1.Equals(0))
            {
                lblRecado1.Text = "Não tivemos nenhuma resposta";
            }
            else if (Estatico.ACERTOSQUEST1.Equals(0) && Estatico.ACERTOSQUEST1 <
Estatico.ERROSQUEST1)
            {
                lblRecado1.Text = "Somente respostas erradas\ntalvez estejam\ncom
dificuldade";
            }
            else if (Estatico.ACERTOSQUEST1 > Estatico.ERROSQUEST1 && Esta-
tico.ERROSQUEST1.Equals(0))
            {
                lblRecado1.Text = "Somente respostas certas \ntalvez acharam\nfa-
cil demais";
            }
            else if (Estatico.ACERTOSQUEST1 > Estatico.ERROSQUEST1)
            {
                lblRecado1.Text = "As pessoas tiveram mais\nfacilidade com esta
questão";
            }
            else if (Estatico.ACERTOSQUEST1 < Estatico.ERROSQUEST1)
            {
                lblRecado1.Text = "As pessoas tiveram mais\ndificuldade com esta
questão";
            }
            else if (Estatico.ACERTOSQUEST1 == Estatico.ERROSQUEST1)
            {
                lblRecado1.Text = "Equilibrio nas respostas";
            }
            else
            {
                lblRecado1.Text = "Algo de errado, não esta certo";
            }
        }
    }
}

```

```

    }

    lblTotalRespostas2.Text += Estatico.TOTALRESPOSTASQUEST2;
    lblTotalAcertos2.Text += Estatico.ACERTOSQUEST2;
    lblTotalErros2.Text += Estatico.ERROSQUEST2;
    if (Estatico.ACERTOSQUEST2.Equals(0) && Estatico.ERROS-
QUEST2.Equals(0))
    {
        lblRecado2.Text = "Não tivemos nenhuma resposta";
    }
    else if (Estatico.ACERTOSQUEST2.Equals(0) && Estatico.ACERTOSQUEST2 <
Estatico.ERROSQUEST2)
    {
        lblRecado2.Text = "Somente respostas erradas\ntalvez estejam\ncom
dificuldade";
    }
    else if (Estatico.ACERTOSQUEST2 > Estatico.ERROSQUEST2 && Esta-
tico.ERROSQUEST2.Equals(0))
    {
        lblRecado2.Text = "Somente respostas certas \ntalvez acharam\nfa-
cil demais";
    }
    else if (Estatico.ACERTOSQUEST2 > Estatico.ERROSQUEST2)
    {
        lblRecado2.Text = "As pessoas tiveram mais\nfacilidade com esta
questão";
    }
    else if (Estatico.ACERTOSQUEST2 < Estatico.ERROSQUEST2)
    {
        lblRecado2.Text = "As pessoas tiveram mais\ndificuldade com esta
questão";
    }

    else if (Estatico.ACERTOSQUEST2 == Estatico.ERROSQUEST2)
    {
        lblRecado2.Text = "Equilibrio nas respostas";
    }
    else
    {
        lblRecado2.Text = "Algo de errado, não esta certo";
    }

    lblTotalRespostas3.Text += Estatico.TOTALRESPOSTASQUEST3;
    lblTotalAcertos3.Text += Estatico.ACERTOSQUEST3;
    lblTotalErros3.Text += Estatico.ERROSQUEST3;
    if (Estatico.ACERTOSQUEST3.Equals(0) && Estatico.ERROS-
QUEST3.Equals(0))
    {
        lblRecado3.Text = "Não tivemos nenhuma resposta";
    }
    else if (Estatico.ACERTOSQUEST3.Equals(0) && Estatico.ACERTOSQUEST3 <
Estatico.ERROSQUEST3)
    {
        lblRecado3.Text = "Somente respostas erradas\ntalvez estejam\ncom
dificuldade";
    }
    else if (Estatico.ACERTOSQUEST3 > Estatico.ERROSQUEST3 && Esta-
tico.ERROSQUEST3.Equals(0))
    {
        lblRecado3.Text = "Somente respostas certas \ntalvez acharam\nfa-
cil demais";
    }
    else if (Estatico.ACERTOSQUEST3 > Estatico.ERROSQUEST3)

```

```

        {
            lblRecado3.Text = "As pessoas tiveram mais\nfacilidade com esta
questão";
        }
        else if (Estatico.ACERTOSQUEST3 < Estatico.ERROSQUEST3)
        {
            lblRecado3.Text = "As pessoas tiveram mais\ndificuldade com esta
questão";
        }

        else if (Estatico.ACERTOSQUEST3 == Estatico.ERROSQUEST3)
        {
            lblRecado3.Text = "Equilibrio nas respostas";
        }
        else
        {
            lblRecado3.Text = "Algo de errado, não esta certo";
        }

        lblTotalRespostas4.Text += Estatico.TOTALRESPOSTASQUEST4;
        lblTotalAcertos4.Text += Estatico.ACERTOSQUEST4;
        lblTotalErros4.Text += Estatico.ERROSQUEST4;
        if (Estatico.ACERTOSQUEST4.Equals(0) && Estatico.ERROS-
QUEST4.Equals(0))
        {
            lblRecado4.Text = "Não tivemos nenhuma resposta";
        }
        else if (Estatico.ACERTOSQUEST4.Equals(0) && Estatico.ACERTOSQUEST4 <
Estatico.ERROSQUEST4)
        {
            lblRecado4.Text = "Somente respostas erradas\ntalvez estejam\ncom
dificuldade";
        }
        else if (Estatico.ACERTOSQUEST4 > Estatico.ERROSQUEST4 && Esta-
tico.ERROSQUEST4.Equals(0))
        {
            lblRecado4.Text = "Somente respostas certas \ntalvez acharam\nfa-
cil demais";
        }
        else if (Estatico.ACERTOSQUEST4 > Estatico.ERROSQUEST4)
        {
            lblRecado4.Text = "As pessoas tiveram mais\nfacilidade com esta
questão";
        }
        else if (Estatico.ACERTOSQUEST4 < Estatico.ERROSQUEST4)
        {
            lblRecado4.Text = "As pessoas tiveram mais\ndificuldade com esta
questão";
        }

        else if (Estatico.ACERTOSQUEST4 == Estatico.ERROSQUEST4)
        {
            lblRecado4.Text = "Equilibrio nas respostas";
        }
        else
        {
            lblRecado4.Text = "Algo de errado, não esta certo";
        }

        lblTotalRespostas5.Text += Estatico.TOTALRESPOSTASQUEST5;
        lblTotalAcertos5.Text += Estatico.ACERTOSQUEST5;
        lblTotalErros5.Text += Estatico.ERROSQUEST5;

```

```

        if (Estatico.ACERTOSQUEST5.Equals(0) && Estatico.ERROS-
QUEST5.Equals(0))
        {
            lblRecado5.Text = "Não tivemos nenhuma resposta";
        }
        else if (Estatico.ACERTOSQUEST5.Equals(0) && Estatico.ACERTOSQUEST5 <
Estatico.ERROSQUEST5)
        {
            lblRecado5.Text = "Somente respostas erradas\ntalvez estejam\ncom
dificuldade";
        }
        else if (Estatico.ACERTOSQUEST5 > Estatico.ERROSQUEST5 && Esta-
tico.ERROSQUEST5.Equals(0))
        {
            lblRecado5.Text = "Somente respostas certas \ntalvez acharam\nta-
cil demais";
        }
        else if (Estatico.ACERTOSQUEST5 > Estatico.ERROSQUEST5)
        {
            lblRecado5.Text = "As pessoas tiveram mais\ntalvez facilidade com esta
questão";
        }
        else if (Estatico.ACERTOSQUEST5 < Estatico.ERROSQUEST5)
        {
            lblRecado5.Text = "As pessoas tiveram mais\ntalvez dificuldade com esta
questão";
        }

        else if (Estatico.ACERTOSQUEST5 == Estatico.ERROSQUEST5)
        {
            lblRecado5.Text = "Equilibrio nas respostas";
        }
        else
        {
            lblRecado5.Text = "Algo de errado, não esta certo";
        }

        lblTotalAcertadas.Text += Estatico.TOTALACERTOS;
        lblTotalErradas.Text += Estatico.TOTALERROS;
    }

    private void pbxBotaoMenu_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}
}

```

APÊNDICE N – GERENCIADORUSUARIOS

```

using SistemaTotemHomemALua.Modelos;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaTotemHomemALua
{
    public partial class GerenciadorUsuarios : Form
    {
        List<Usuarios> usuarios = new List<Usuarios>();

        public GerenciadorUsuarios()
        {
            InitializeComponent();
            dgvDev.DataSource = usuarios;

            //Codigo para preencher a tabela para a apresentação
            if (File.Exists(@"C:\LIT\BancoDeDados\usuarios.json"))
            {
                JsonControle jsonControle = new JsonControle();
                usuarios = jsonControle.DeserializarLista(@"C:\LIT\BancoDeDa-
dos\usuarios.json");
                dgvDev.DataSource = null;
                dgvDev.DataSource = usuarios;
            }

            public void AdicionarPessoaPreencherTabela(string nome, int idade, int
questionarioNota, string comentario)
            {
                usuarios.Add(new Usuarios() { Nome = nome, Idade = idade, Questio-
narioNota = questionarioNota, Comentario = comentario });

                usuarios = usuarios.OrderByDescending(u => u.Questionario-
Nota).ToList();
                for (int i = 0; i < usuarios.Count; i++)
                {
                    usuarios[i].Posicao = i;
                }

                dgvDev.DataSource = null;
                dgvDev.DataSource = usuarios;
            }

            public void AdicionarPessoa(string mensagemForm1, string nome, int idade,
int questionarioNota, string comentario)
            {
                if (mensagemForm1.Equals("Armazenar"))
                {
                    usuarios.Add(new Usuarios() { Nome = nome, Idade = idade, Questio-
narioNota = questionarioNota, Comentario = comentario });
                }
            }
        }
    }
}

```

```

        ArmazenarEstatisticas();

        usuarios = usuarios.OrderByDescending(u => u.Questionario-
Nota).ToList();
        for (int i = 0; i < usuarios.Count; i++)
        {
            usuarios[i].Posicao = i;
        }

        dgvDev.DataSource = null;
        dgvDev.DataSource = usuarios;
    }
}

public void ArmazenarEstatisticas()
{
    //Armazenar diferentes notas de avaliação ao museu

    Estatico.TOTALRESPOSTASQUEST1 += Estatico.TEMPACERTOSQUEST1 + Esta-
tico.TEMPERROSQUEST1;
    Estatico.TOTALRESPOSTASQUEST2 += Estatico.TEMPACERTOSQUEST2 + Esta-
tico.TEMPERROSQUEST2;
    Estatico.TOTALRESPOSTASQUEST3 += Estatico.TEMPACERTOSQUEST3 + Esta-
tico.TEMPERROSQUEST3;
    Estatico.TOTALRESPOSTASQUEST4 += Estatico.TEMPACERTOSQUEST4 + Esta-
tico.TEMPERROSQUEST4;
    Estatico.TOTALRESPOSTASQUEST5 += Estatico.TEMPACERTOSQUEST5 + Esta-
tico.TEMPERROSQUEST5;

    Estatico.ACERTOSQUEST1 += Estatico.TEMPACERTOSQUEST1;
    Estatico.ACERTOSQUEST2 += Estatico.TEMPACERTOSQUEST2;
    Estatico.ACERTOSQUEST3 += Estatico.TEMPACERTOSQUEST3;
    Estatico.ACERTOSQUEST4 += Estatico.TEMPACERTOSQUEST4;
    Estatico.ACERTOSQUEST5 += Estatico.TEMPACERTOSQUEST5;

    Estatico.ERROSQUEST1 += Estatico.TEMPERROSQUEST1;
    Estatico.ERROSQUEST2 += Estatico.TEMPERROSQUEST2;
    Estatico.ERROSQUEST3 += Estatico.TEMPERROSQUEST3;
    Estatico.ERROSQUEST4 += Estatico.TEMPERROSQUEST4;
    Estatico.ERROSQUEST5 += Estatico.TEMPERROSQUEST5;

    Estatico.TOTALERROS = Estatico.ERROSQUEST1 + Estatico.ERROSQUEST2 +
Estatico.ERROSQUEST3 + Estatico.ERROSQUEST4 + Estatico.ERROSQUEST5;
    Estatico.TOTALACERTOS = Estatico.ACERTOSQUEST1 + Estatico.ACERTOS-
QUEST2 + Estatico.ACERTOSQUEST3 + Estatico.ACERTOSQUEST4 + Estatico.ACERTOS-
QUEST5;
}

public void SalvarSerializar(string caminhoDiretorio, string caminhoAr-
quivo)
{
    JsonControle jsonControle = new JsonControle();
    jsonControle.SerializarLista(usuarios, caminhoDiretorio, caminhoAr-
quivo);
}
}
}

```

APÊNDICE O – CONTROLE

```

namespace SistemaTotemHomemALua.Modelos
{
    internal class Controle
    {
        private string mensagem = "";
        private string mensagemForm1 = "";

        public void ValidarResposta(int idPergunta, string Resposta)
        {
            Validacao validacao = new Validacao();
            int i = validacao.ValidarResposta(idPergunta, Resposta);
            switch (i)
            {
                case 0:
                    switch (idPergunta)
                    {
                        case 1:
                            Estatico.TEMPERROSQUEST1 = 1;
                            break;

                        case 2:
                            Estatico.TEMPERROSQUEST2 = 1;
                            break;

                        case 3:
                            Estatico.TEMPERROSQUEST3 = 1;
                            break;

                        case 4:
                            Estatico.TEMPERROSQUEST4 = 1;
                            break;

                        case 5:
                            Estatico.TEMPERROSQUEST5 = 1;
                            break;
                    }
                    break;

                case 1:
                    Estatico.QUESTIONARIOUOTAUSUARIO += i;
                    switch (idPergunta)
                    {
                        case 1:
                            Estatico.TEMPACERTOSQUEST1 = 1;
                            break;

                        case 2:
                            Estatico.TEMPACERTOSQUEST2 = 1;
                            break;

                        case 3:
                            Estatico.TEMPACERTOSQUEST3 = 1;
                            break;

                        case 4:
                            Estatico.TEMPACERTOSQUEST4 = 1;
                            break;

                        case 5:
                            Estatico.TEMPACERTOSQUEST5 = 1;
                    }
                }
            }
        }
    }
}

```



```

                break;
            }
            break;
        }
    }

    public void ArmazenarNomeIdade(string usuarioNome, string usuarioIdade)
    {
        mensagem = "";
        Validacao validacao = new Validacao();
        validacao.ValidarNome(usuarioNome);
        if (!validacao.Mensagem.Equals(""))
        {
            mensagem = validacao.Mensagem;
        }
        else
        {
            validacao.ValidarIdade(usuarioIdade);
            if (!validacao.Mensagem.Equals(""))
            {
                mensagem = validacao.Mensagem;
            }
        }
    }

    public void ArmazenarAvaliacaoComentario(int notaQualidadeObras, int notaVisitaMuseu, int notaRecomendacao, string usuarioComentario)
    {
        Estatico.SOMATOTALQUALIDADEOBRAS += notaQualidadeObras;
        Estatico.NUMEROAVALIACOESQUALIDADEOBRAS++;

        Estatico.SOMATOTALVISITAMUSEU += notaVisitaMuseu;
        Estatico.NUMEROAVALIACOESVISITAMUSEU++;

        Estatico.SOMATOTALRECOMENDACAO += notaRecomendacao;
        Estatico.NUMEROAVALIACOESRECOMENDACAO++;

        Estatico.COMENTARIOUSUARIO = usuarioComentario;
    }

    public void ConcluirCadastro(int notaQualidadeObras, int notaVisitaMuseu, int notaRecomendacao, string usuarioComentario)
    {
        ArmazenarAvaliacaoComentario(notaQualidadeObras, notaVisitaMuseu, notaRecomendacao, usuarioComentario);
        mensagemForm1 = "Armazenar";
    }

    public string Teclado(int KeyID)
    {
        switch (KeyID)
        {
            #region Letras Maiusculas
            case 1: return "A"; break;
            case 2: return "B"; break;
            case 3: return "C"; break;
            case 4: return "D"; break;
            case 5: return "E"; break;
            case 6: return "F"; break;

```

```

case 7: return "G"; break;
case 8: return "H"; break;
case 9: return "I"; break;
case 10: return "J"; break;
case 11: return "K"; break;
case 12: return "L"; break;
case 13: return "M"; break;
case 14: return "N"; break;
case 15: return "O"; break;
case 16: return "P"; break;
case 17: return "Q"; break;
case 18: return "R"; break;
case 19: return "S"; break;
case 20: return "T"; break;
case 21: return "U"; break;
case 22: return "V"; break;
case 23: return "W"; break;
case 24: return "X"; break;
case 25: return "Y"; break;
case 26: return "Z"; break;
#endregion

```

```

#region Letras Minusculas
case 27: return "a"; break;
case 28: return "b"; break;
case 29: return "c"; break;
case 30: return "d"; break;
case 31: return "e"; break;
case 32: return "f"; break;
case 33: return "g"; break;
case 34: return "h"; break;
case 35: return "i"; break;
case 36: return "j"; break;
case 37: return "k"; break;
case 38: return "l"; break;
case 39: return "m"; break;
case 40: return "n"; break;
case 41: return "o"; break;
case 42: return "p"; break;
case 43: return "q"; break;
case 44: return "r"; break;
case 45: return "s"; break;
case 46: return "t"; break;
case 47: return "u"; break;
case 48: return "v"; break;
case 49: return "w"; break;
case 50: return "x"; break;
case 51: return "y"; break;
case 52: return "z"; break;
#endregion

```

```

#region Teclas Especiais
case 53: return ","; break;
case 54: return "."; break;
case 55: return " "; break;
case 56: return "Ç"; break;
case 57: return "ç"; break;
#endregion

```

```

#region Numeros
case 58: return "1"; break;
case 59: return "2"; break;
case 60: return "3"; break;

```

```
        case 61: return "4"; break;
        case 62: return "5"; break;
        case 63: return "6"; break;
        case 64: return "7"; break;
        case 65: return "8"; break;
        case 66: return "9"; break;
        case 67: return "0"; break;
        #endregion

        default: return " "; break;
    }

    return " ";
}

public string Mensagem { get => mensagem; }
public string MensagemForm1 { get => mensagemForm1; }
}

}
```

APÊNDICE P – VALIDACAO

```

namespace SistemaTotemHomemALua.Modelos
{
    internal class Validacao
    {
        private string mensagem = "";

        public int ValidarResposta(int idPergunta, string Resposta)
        {
            switch (idPergunta)
            {
                case 1:
                    Estatico.TEMPRESPOSTAQUEST1 = Resposta;
                    if (Resposta.Equals("D) 1969"))
                    {
                        return 1;
                    }
                    break;

                case 2:
                    Estatico.TEMPRESPOSTAQUEST2 = Resposta;
                    if (Resposta.Equals("C) Saturn V"))
                    {
                        return 1;
                    }
                    break;

                case 3:
                    Estatico.TEMPRESPOSTAQUEST3 = Resposta;
                    if (Resposta.Equals("A) Edwin E. Aldrin Jr."))
                    {
                        return 1;
                    }
                    break;

                case 4:
                    Estatico.TEMPRESPOSTAQUEST4 = Resposta;
                    if (Resposta.Equals("B) Early Apollo Scientific Experiments
Package"))
                    {
                        return 1;
                    }
                    break;

                case 5:
                    Estatico.TEMPRESPOSTAQUEST5 = Resposta;
                    if (Resposta.Equals("C) Michael Collins"))
                    {
                        return 1;
                    }
                    break;

                default:
                    break;
            }
        }
    }
}

```

```

    }

    return 0;
}

public void ValidarNome(string nome)
{
    mensagem = "";
    if (!nome.Equals(""))
    {
        Estatico.NOMEUSUARIO = nome;
    }
    else
    {
        mensagem = "Inserir Nome";
    }
}

public void ValidarIdade(string idade)
{
    mensagem = "";
    if (!idade.Equals(""))
    {
        try
        {
            Estatico.IDADEUSUARIO = Convert.ToInt32(idade);
        }
        catch (Exception e)
        {
            mensagem = "Inserir Somente Numero";
        }
    }
    else
    {
        mensagem = "Inserir Idade";
    }
}

public string Mensagem { get => mensagem; }
}
}

```

APÊNDICE Q – ESTATICO

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SIsistemaTotemHomemALua.Modelos
{
    internal static class Estatico
    {
        public static string NOMEUSUARIO = "";
        public static int IDADEUSUARIO = 0;
        public static int QUESTIONARIOUOTATUSUARIO = 0;

        public static int SOMATOTALQUALIDADEOBRAS = 0;
        public static int NUMEROAVALIACOESQUALIDADEOBRAS = 0;
        public static int SOMATOTALVISITAMUSEU = 0;
        public static int NUMEROAVALIACOESVISITAMUSEU = 0;
        public static int SOMATOTALRECOMENDACAO = 0;
        public static int NUMEROAVALIACOESRECOMENDACAO = 0;
        public static string COMENTARIOUSUARIO = "";

        public static int TOTALRESPOSTASQUEST1 = 0;
        public static int TOTALRESPOSTASQUEST2 = 0;
        public static int TOTALRESPOSTASQUEST3 = 0;
        public static int TOTALRESPOSTASQUEST4 = 0;
        public static int TOTALRESPOSTASQUEST5 = 0;

        public static int ACERTOSQUEST1 = 0;
        public static int ACERTOSQUEST2 = 0;
        public static int ACERTOSQUEST3 = 0;
        public static int ACERTOSQUEST4 = 0;
        public static int ACERTOSQUEST5 = 0;

        public static int ERROSQUEST1 = 0;
        public static int ERROSQUEST2 = 0;
        public static int ERROSQUEST3 = 0;
        public static int ERROSQUEST4 = 0;
        public static int ERROSQUEST5 = 0;

        public static int TOTALERROS = 0;
        public static int TOTALACERTOS = 0;

        public static string TEMPRESPOSTAQUEST1 = "";
        public static string TEMPRESPOSTAQUEST2 = "";
        public static string TEMPRESPOSTAQUEST3 = "";
        public static string TEMPRESPOSTAQUEST4 = "";
        public static string TEMPRESPOSTAQUEST5 = "";

        public static int TEMPACERTOSQUEST1 = 0;
        public static int TEMPACERTOSQUEST2 = 0;
        public static int TEMPACERTOSQUEST3 = 0;
        public static int TEMPACERTOSQUEST4 = 0;
        public static int TEMPACERTOSQUEST5 = 0;

        public static int TEMPERROSQUEST1 = 0;
        public static int TEMPERROSQUEST2 = 0;
        public static int TEMPERROSQUEST3 = 0;
        public static int TEMPERROSQUEST4 = 0;
        public static int TEMPERROSQUEST5 = 0;
    }
}

```

} }

APÊNDICE R – JSONCONTROLE

```

using Newtonsoft.Json;

namespace SistemaTotemHomemALua.Modelos
{
    internal sealed class JsonControle
    {

        public List<Usuarios> DeserializarLista(string caminhoArquivo)
        {
            string json = File.ReadAllText(caminhoArquivo);
            return JsonConvert.DeserializeObject<List<Usuarios>>(json);
            return null;
        }

        public void SerializarLista(List<Usuarios> usuarios, string caminhoDire-
torio, string caminhoArquivo)
        {
            if (Directory.Exists(caminhoDiretorio))
            {
                try
                {
                    var strJson = JsonConvert.SerializeObject(usuarios, Format-
ting.Indented);
                    SalvarArquivo(strJson, caminhoArquivo);
                }
                catch (Exception e)
                {
                    MessageBox.Show("Erro na função de SerializarLista");
                }
            }
            else
            {
                Directory.CreateDirectory(caminhoDiretorio);
                SerializarLista(usuarios, caminhoDiretorio, caminhoArquivo);
            }
        }

        public void SalvarArquivo(string strJson, string caminhoArquivo)
        {
            try
            {
                File.WriteAllText(caminhoArquivo, strJson);
            }
            catch (Exception e)
            {
                MessageBox.Show("Erro na função SalvarArquivo");
            }
        }
    }
}








```


APÊNDICE S – USUARIOS

```
namespace SistemaTotemHomemALua.Modelos
{
    internal class Usuarios
    {
        public int Posicao { get; set; }
        public string Nome { get; set; }
        public int Idade { get; set; }
        public int QuestionarioNota { get; set; }
        public string Comentario { get; set; }
    }
}
```

APÊNDICE T – ARQUIVOS DO SISTEMA

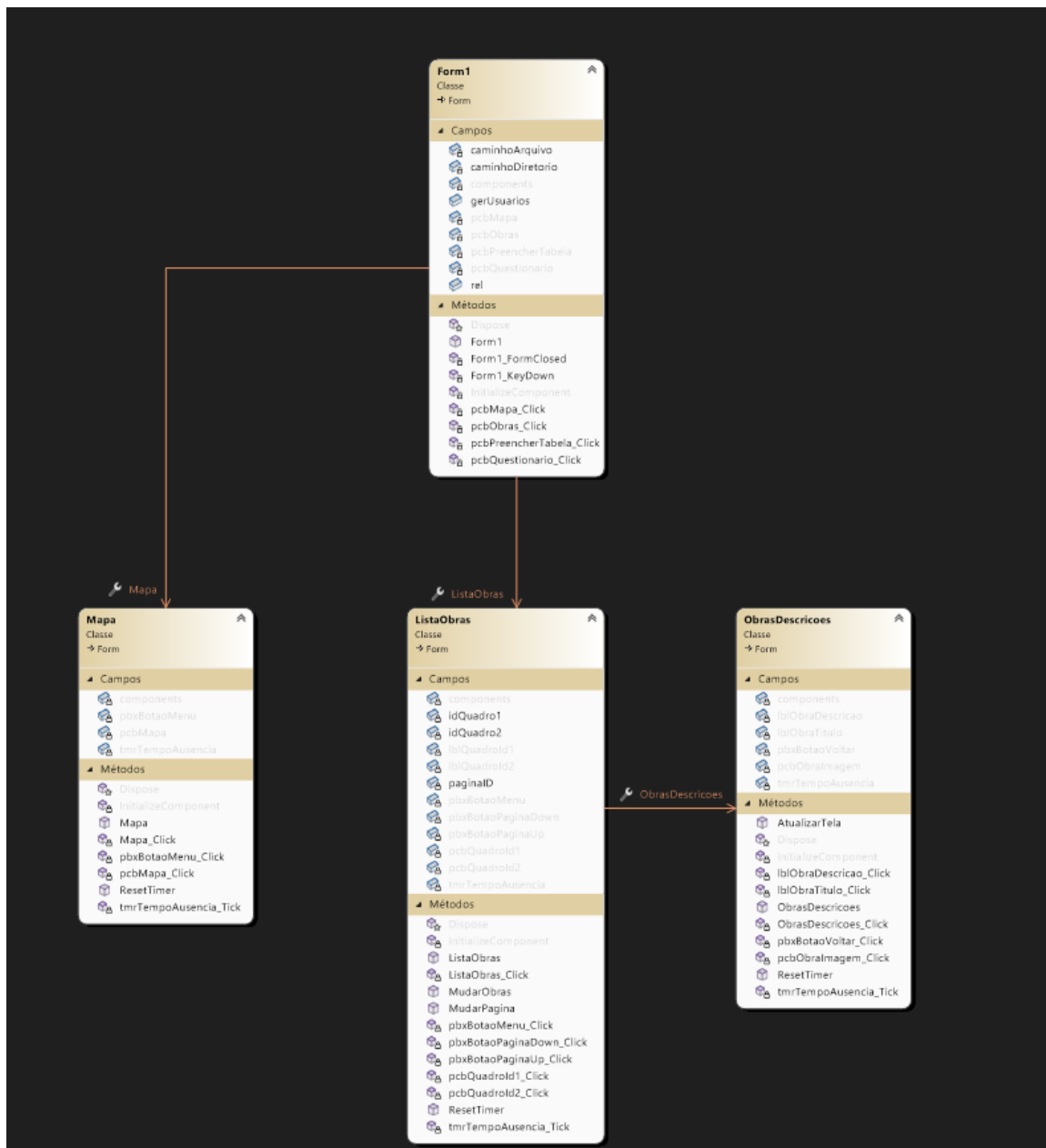
Figura 34: Arquivos do Sistema

	Imagens	22/05/2024 11:44	Pasta de arquivos	
	Newtonsoft.Json.dll	08/03/2023 04:09	Extensão de aplica...	696 KB
	SlstemaTotemHomemALua.deps.json	19/05/2024 15:13	Arquivo Fonte JSON	2 KB
	SlstemaTotemHomemALua.dll	22/05/2024 11:44	Extensão de aplica...	10.965 KB
	SlstemaTotemHomemALua.exe	22/05/2024 11:44	Aplicativo	140 KB
	SlstemaTotemHomemALua.pdb	22/05/2024 11:44	Arquivo PDB	56 KB
	SlstemaTotemHomemALua.runtimeconfi...	15/05/2024 01:02	Arquivo Fonte JSON	1 KB

Fonte: Autoria Própria

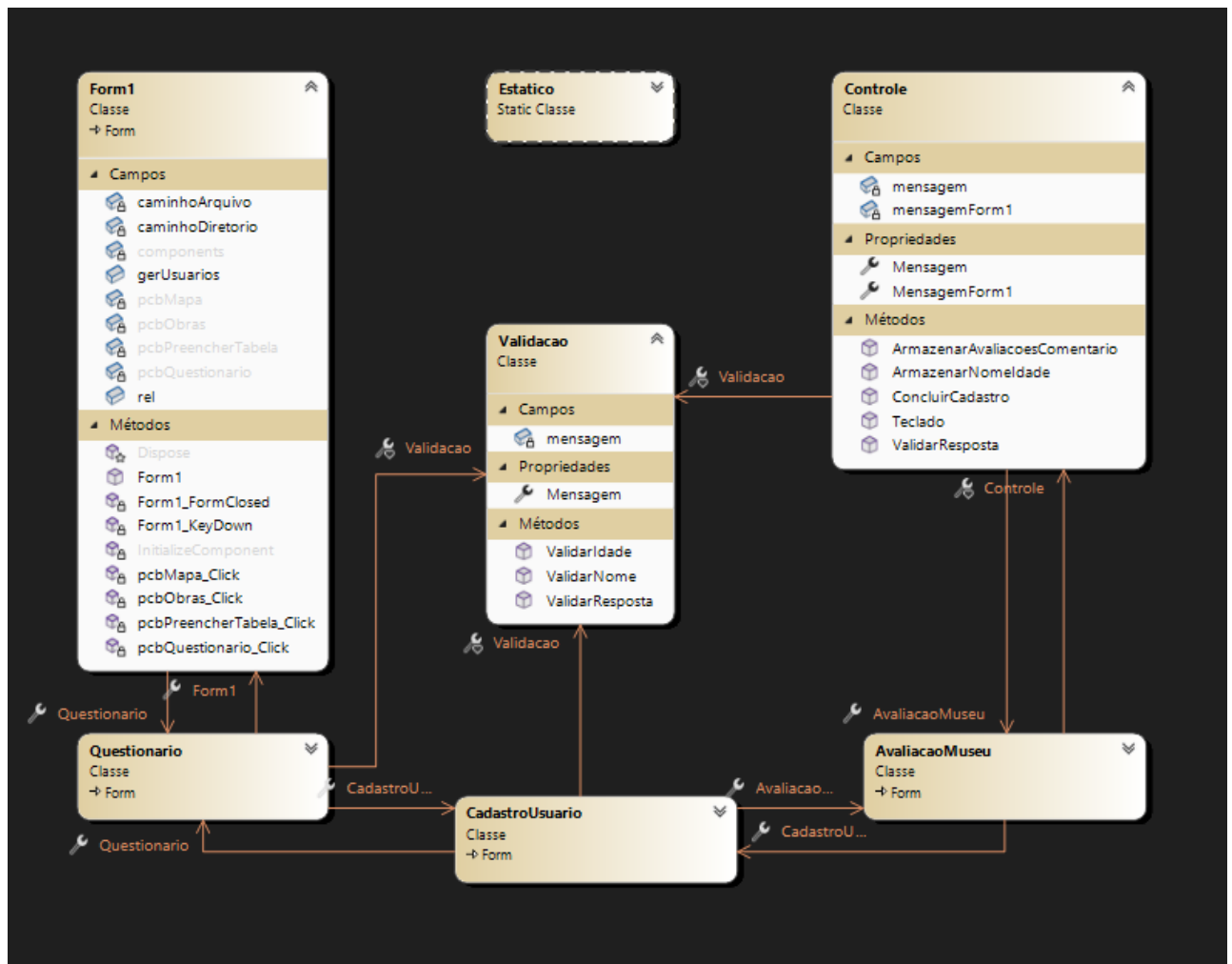
APÊNDICE U – DIAGRAMAS DE CLASSE

Figura 35: Diagrama de Classe - Mapa e ListaObras



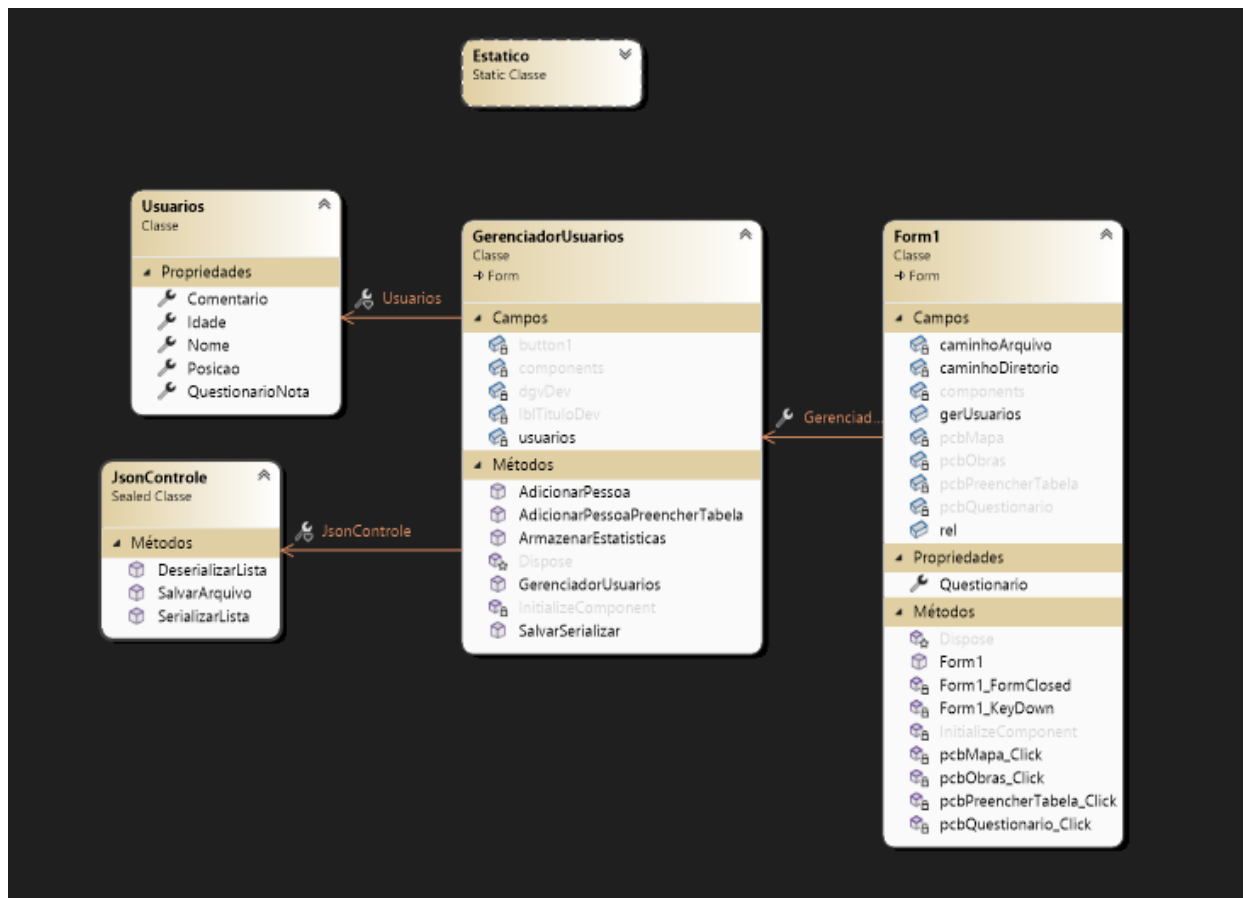
Fonte: Autoria Própria

Figura 36: Diagrama de Classe - Sequência Questionário



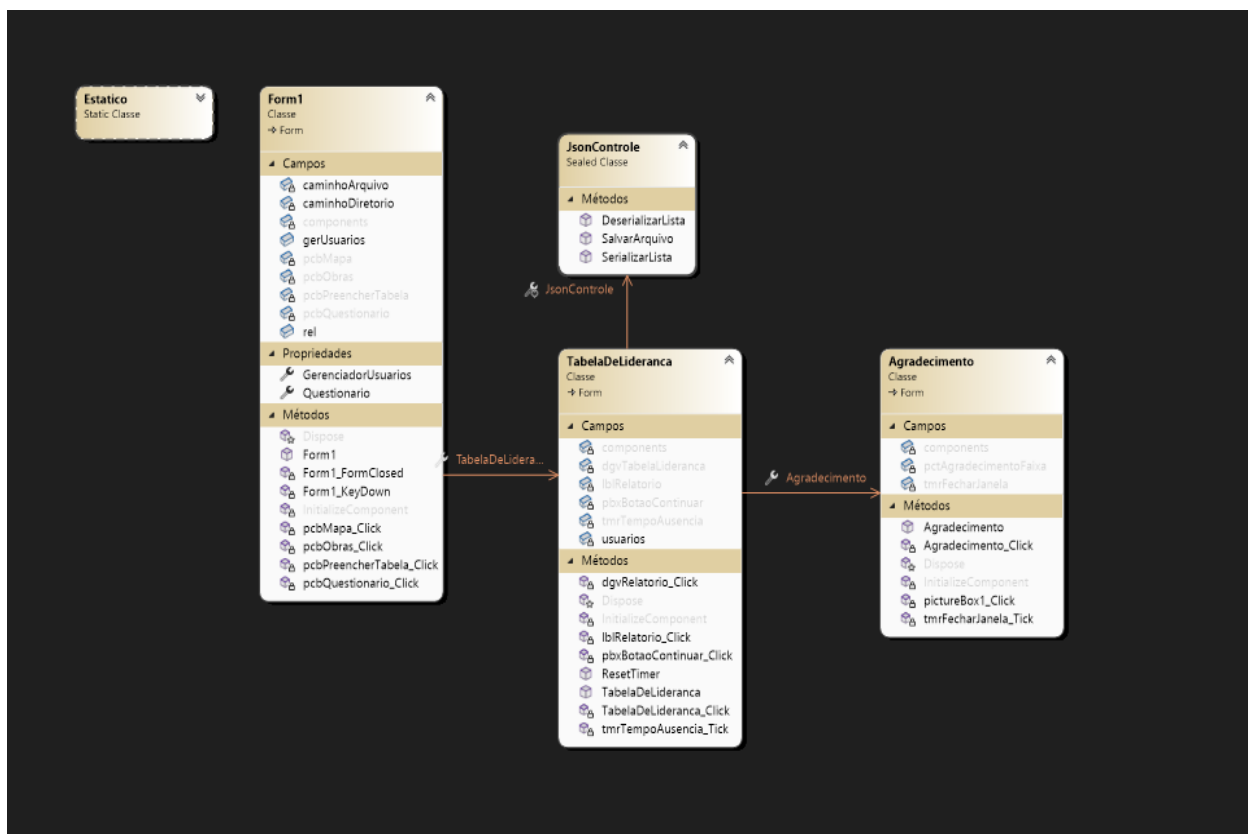
Fonte: Autoria Própria

Figura 37: Diagrama de Classe - Salvar Usuário



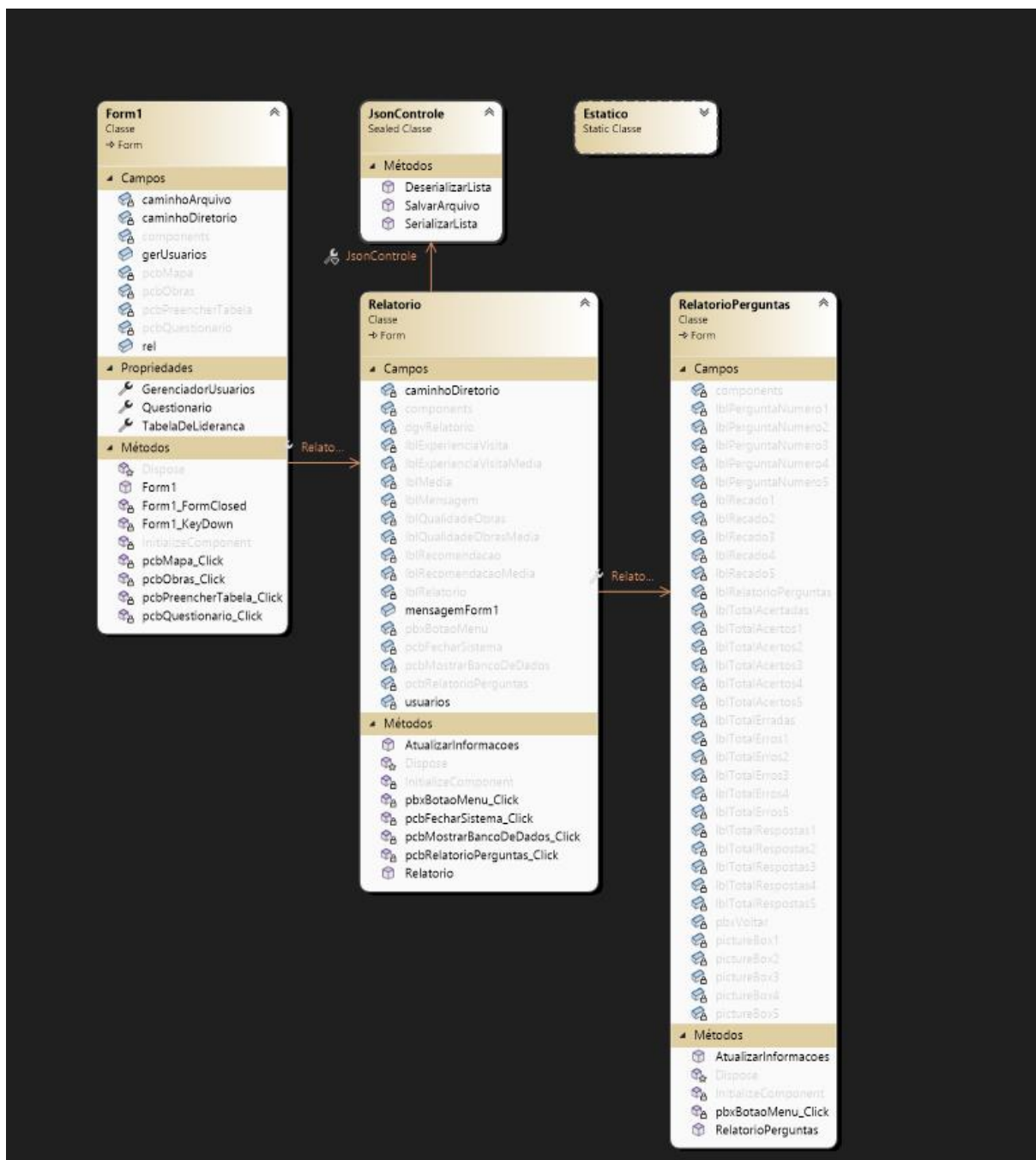
Fonte: Autoria Própria

Figura 38: Diagrama de Classe - Sequência Final do Questionario/ Deserialização



Fonte: Autoria Própria

Figura 39: Diagrama de Classe - Relatório



Fonte: Autoria Própria

ANEXO A - OBRAS E DESCRIÇÕES

Figura 40: Obra Astronauta Neil Armstrong



Fonte: NASA

“Neil Armstrong, vestido com seu traje espacial, posa para seu retrato oficial da Apollo 11.” (NASA, 2024).

Figura 41: Obra Astronauta Edwin E. Aldrin Jr.



Fonte: NASA

“Astronauta Edwin E. Aldrin Jr. Aldrin foi piloto do módulo lunar da missão de pouso lunar Apollo 11.” (NASA, 2024).

Figura 42: Obra Astronauta Michael Collins



Fonte: NASA

“Collins renunciou ao programa espacial da NASA desde que esta foto foi tirada.” (NASA, 2024).

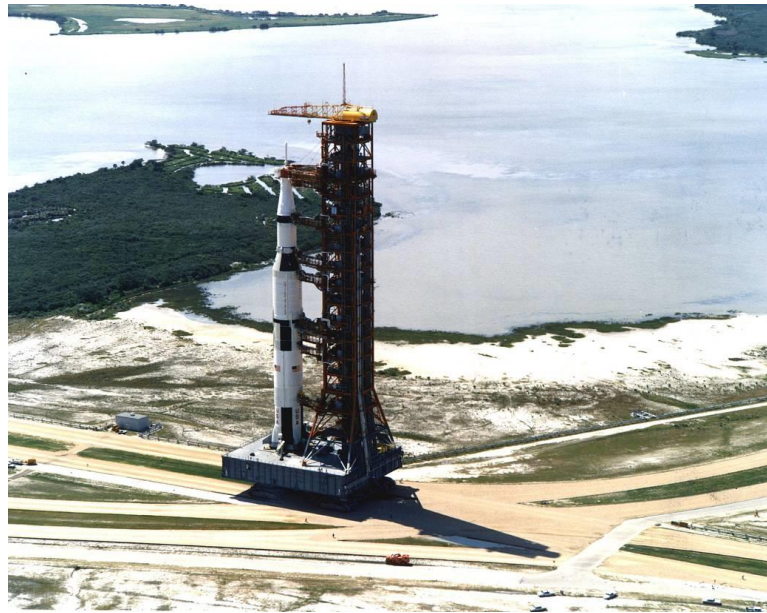
Figura 43: Obra A tripulação da Apollo 11



Fonte: NASA

“A tripulação da Apollo 11 realiza uma verificação funcional e de ajuste do compartimento da tripulação, dos equipamentos e locais de armazenamento, em seu módulo de comando.” (NASA, 2024).

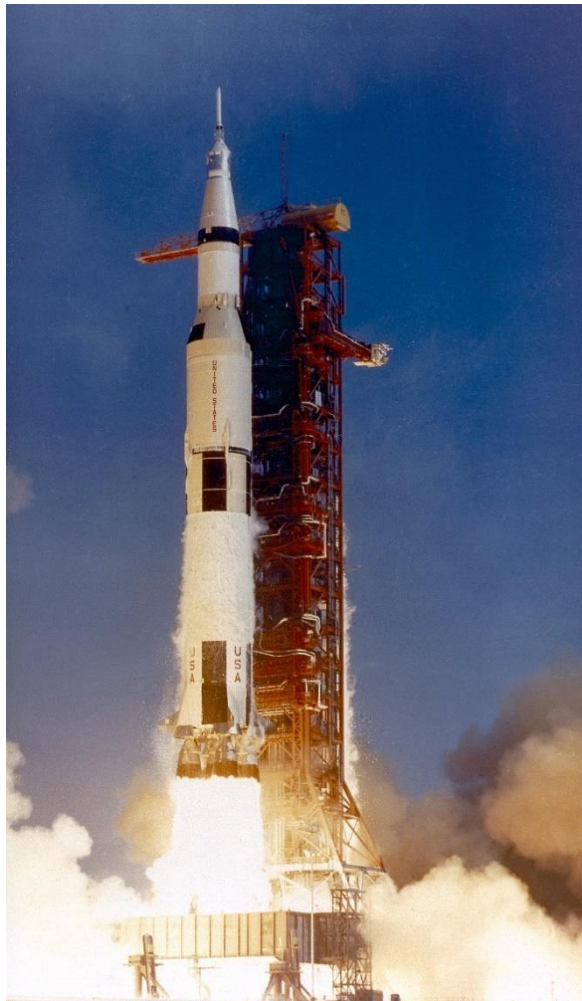
Figura 44: Obra Saturn V Ventures



Fonte: NASA

“Carregando o foguete Apollo 11 Saturn V e o lançador móvel, o rastreador avança lentamente ao longo da jornada de cinco quilômetros até a plataforma de lançamento 39A.” (NASA, 2024).

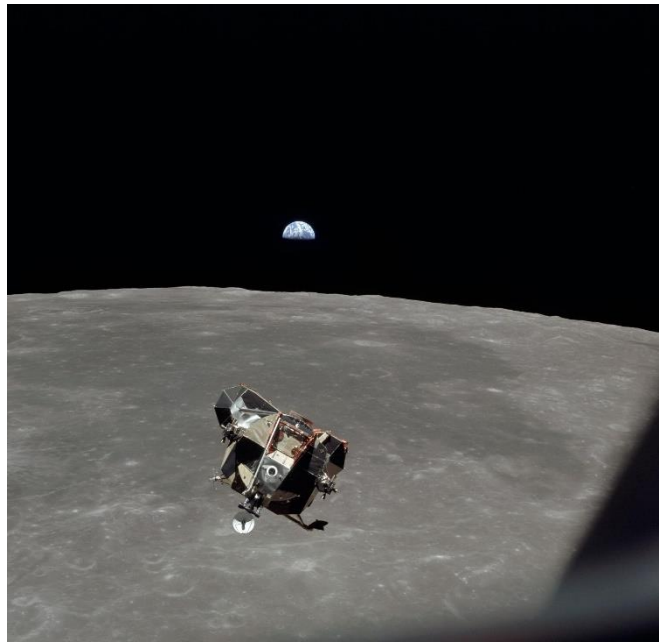
Figura 45: Obra Programa Saturno Apollo



Fonte: NASA

“Esta fotografia mostra o veículo de lançamento Saturn V (SA-506) para a decolagem da missão Apollo 11 às 8h32 CDT, 16 de julho de 1969, do complexo de lançamento 39A no Centro Espacial Kennedy.” (NASA, 2024).

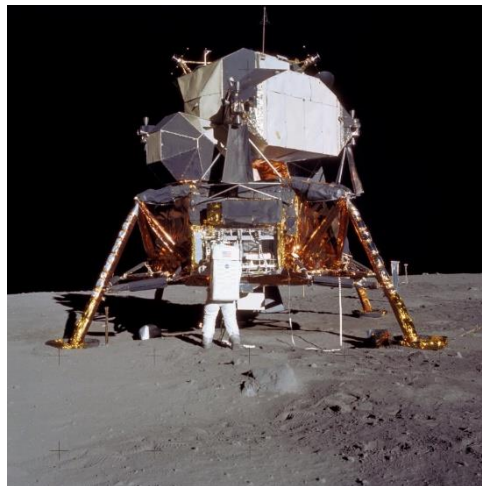
Figura 46: Obra Módulo Lunar Apollo 11



Fonte: NASA

“O estágio de subida do Módulo Lunar da Apollo 11, com os astronautas Neil A. Armstrong e Edwin E. Aldrin Jr. a bordo, é fotografado dos Módulos de Comando e Serviço (CSM) durante o encontro na órbita lunar.” (NASA, 2024).

Figura 47: Obra Módulo Lunar da Apollo 11 na superfície lunar



Fonte: NASA

“O astronauta Edwin E. Aldrin Jr., piloto do módulo lunar, se prepara para implantar o Early Apollo Scientific Experiments Package (EASEP) durante a atividade extraveicular da superfície lunar (EVA) da Apollo 11.” (NASA, 2024).

Figura 48: Obra Pegada no solo lunar



Fonte: NASA

“Uma visão aproximada da bota e da pegada de um astronauta no solo lunar, fotografada com uma câmera de superfície lunar de 70 mm durante a atividade extraveicular da superfície lunar (EVA) da Apollo 11.” (NASA, 2024)

Figura 49: Obra Vista da lua



Fonte: NASA

“Esta vista da espaçonave Apollo 11 mostra a Terra elevando-se acima do horizonte da lua. O terreno lunar retratado está na área do Mar de Smyth, no lado mais próximo.” (NASA, 2024).