



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS TERESINA ZONA SUL
CURSO DE BACHARELADO EM ENGENHARIA CIVIL**

ANA PAULA GOMES SOARES

**DESENVOLVIMENTO DE APLICATIVO LOW-CODE BASEADO NA
METODOLOGIA FMEA PARA O AUXÍLIO NAS ATIVIDADES DE INSPEÇÃO
PREDIAL E CLASSIFICAÇÃO DE RISCO**

TERESINA

2025

ANA PAULA GOMES SOARES

DESENVOLVIMENTO DE APLICATIVO LOW-CODE BASEADO NA METODOLOGIA
FMEA PARA O AUXÍLIO NAS ATIVIDADES DE INSPEÇÃO PREDIAL E
CLASSIFICAÇÃO DE RISCO

Trabalho de Conclusão de Curso (monografia)
apresentado como exigência parcial para obtenção
do diploma do Curso de Engenharia Civil do
Instituto Federal de Educação, Ciência e Tecnologia
do Piauí, Campus Zona Sul.

Orientador: Prof. Dr. Helder Pontes Gomes

TERESINA

2025

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

Soares, Ana Paula Gomes

S676d Desenvolvimento de aplicativo Low-Code baseado na metodologia FMEA para o auxílio nas atividades de inspeção predial e classificação de risco / Ana Paula Gomes Soares. - 2025.
112 f.: il. color.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia Civil) - Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Zona Sul, 2025.

Orientador : Prof Dr. Helder Pontes Gomes.

1. Inspeção predial. 2. Metodologia FMEA. 3. Aplicativo móvel. I.Título.

CDD - 624

ANA PAULA GOMES SOARES

DESENVOLVIMENTO DE APLICATIVO LOW-CODE BASEADO NA METODOLOGIA
FMEA PARA O AUXÍLIO NAS ATIVIDADES DE INSPEÇÃO PREDIAL E
CLASSIFICAÇÃO DE RISCO

Trabalho de Conclusão de Curso (monografia)
apresentado como exigência parcial para obtenção
do diploma do Curso de Engenharia Civil do
Instituto Federal de Educação, Ciência e Tecnologia
do Piauí, Campus Teresina Zona Sul.

Aprovada em: ____/____/____.

BANCA EXAMINADORA

Prof. Dr. Helder Pontes Gomes (Orientador)
Instituto Federal do Piauí (IFPI)

Prof. Dr. Fernando José Guimarães Ferreira
Instituto Federal do Piauí (IFPI)

Prof. Dr. Wallison Angelim Medeiros
Instituto Federal do Piauí (IFPI)

Dedico este trabalho à minha família, em especial aos meus pais,
Aos amigos e mestres.

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus por me guiar na minha jornada acadêmica, me concedendo o desejo pela vida e mostrando que deve ser vivida em sua plenitude.

Agradeço aos meus pais. À minha mãe, Rita, por acreditar no meu potencial e sempre incentivar meus estudos. Obrigada por todo seu amor incondicional, por sua determinação em buscar recursos para que meus sonhos se tornassem realidade. Obrigada por ser minha inspiração e me tornar uma pessoa forte e determinada. Agradeço ao meu pai, Evandro, que apesar da distância, sempre mostrou seu apoio à minha educação, obrigada por todo seu amor e carinho. Obrigada por serem meu alicerce.

Aos meus avós Aldenora, José Benedito (*in memoriam*), Maria e Nagibe (*in memoriam*), sou grata por todo o carinho, amor e cuidado que sempre me dedicaram.

Aos meus tios Maria do Desterro, Natal, Antônia de Fátima, Joelson, Ana Paula, Maria Luísa e Francisco de Assis, agradeço imensamente pelo acolhimento, apoio e afeto incondicional que sempre me ofereceram.

Aos meus primos, Natália e Nataniel, que foram meus companheiros por 2 anos em Teresina, muito obrigada por estarem ao meu lado, pela companhia e por me proporcionarem momentos de alegria e conhecer pessoas boas como a Maria Clara.

Aos meus primos, Sheylane, Shirley, Sheyla e Victor, agradeço por serem meus irmãos mais velhos e pela amizade, sempre compartilhando bons momentos e boas risadas. À minha “irmã” mais velha, Shirley, minha inspiração acadêmica, obrigada por todo apoio educacional e incentivo, você foi crucial para essa jornada. Agradeço também ao Denis, por todas as dicas para tornar possível o desenvolvimento da ideia deste trabalho.

Agradeço a todos os professores do curso pelos ensinamentos e empenho que contribuíram de maneira significativa para minha formação acadêmica. Em especial, ao Prof. Dr. Ailton Soares, pela sua excelência e dedicação durante e após a coordenação do curso de Engenharia Civil.

Ao meu orientador, Prof. Dr. Helder Pontes, agradeço por todos os conselhos e conhecimentos compartilhados. Obrigada pela disponibilidade, paciência e apoio durante o período de orientação para o desenvolvimento deste trabalho.

Agradeço ao professor Fernando Guimarães, pela dedicação e clareza ao transmitir sua experiência, que despertaram em mim o interesse e o entusiasmo pela

área de estrutura.

Agradeço aos meus amigos de graduação, Ismara, Tiago, Alberto e July, por compartilharem bons momentos nessa jornada acadêmica e profissional, em especial ao grupo PALMA, composto por Pedro, Angelo, Lysia e Marcos, pela amizade e companheirismo. À minha dupla e companheira de graduação, Lysia, obrigada pela convivência e pelos momentos de fofocas e alegria compartilhadas.

Por fim, agradeço a todos que contribuíram e torceram pela realização deste sonho.

“Nada na vida deve ser temido, somente compreendido.
Agora é hora de compreender mais para temer menos.”

Marie Curie

RESUMO

A inspeção predial é um processo fundamental para avaliar as condições de utilização da estrutura. Essa atividade identifica as manifestações patológicas que podem comprometer o desempenho e a segurança do edifício, além de sugerir intervenções, especificando as ações mais urgentes com base na extensão e impacto, analisadas via metodologias de gestão de riscos. A aplicação de ferramentas de gestão de riscos nas atividades de inspeções prediais torna a tomada de decisão mais eficiente e embasada. No entanto, esse processo pode ser lento, principalmente quando há um grande volume de informações a serem gerenciadas. Nesse sentido, este trabalho busca desenvolver um aplicativo *low-code* baseado na metodologia de gestão de riscos FMEA (Análise dos Modos e Efeitos das Falhas), com a finalidade de organizar e agilizar o processamento das informações coletadas durante as inspeções prediais, desenvolvendo uma ferramenta prática e eficiente para profissionais de engenharia e arquitetura. Por meio da plataforma FlutterFlow, foi possível criar o aplicativo MAniFE, que permite a inserção de dados e que através de critérios de severidade, ocorrência e detecção, hierarquiza as manifestações patológicas, auxiliando na definição de prioridades de intervenção e na sintetização das informações em um documento PDF. Além disso, facilita o monitoramento da edificação e avaliação da eficácia das ações corretivas que podem ser consultadas tanto no aplicativo quanto no banco de dados.

Palavras-chave: inspeção predial; metodologia FMEA; aplicativo móvel.

ABSTRACT

Building inspection is a fundamental process for evaluating the operational condition of the structure. This activity identifies the pathological manifestations that may compromise the functionality and safety of the building, in addition to suggesting interventions, specifying the most urgent actions based on the extent and impact, analyzed by the risk management methodologies. The risk management tools in building inspection activities makes decision-making more efficient and well-founded. However, this process is slow, especially when there is a large volume of information to be managed. This work seeks to develop a low-code application based on the FMEA risk management methodology (Analysis of Failure Modes and Effects), in order to organize and speed up the processing of information collected during building inspections, providing a practical and efficient tool for professionals in engineering and architecture. The FlutterFlow platform was used to create the MAniFE application, which allows the insertion of data and by the criteria of severity, occurrence and detection, hierarchizes pathological manifestations, assisting the definition of intervention priorities and the synthesis of information in a PDF document. In addition, it facilitates the monitoring of the building and evaluation of the effectiveness of corrective actions that can be consulted both in the application and in the database.

Keywords: building inspection; FMEA methodology; mobile application.

LISTA DE ILUSTRAÇÕES

Figura 1 - Etapas do desenvolvimento da atividade de Inspeção Predial	21
Figura 2 - Distribuição da incidência dos acidentes prediais por origem	26
Figura 3 - Classificação da FMEA.....	29
Figura 4 - Relação Causa-Modo-Efeito	31
Figura 5 - Fluxograma de execução da FMEA.....	32
Figura 6 - Fluxograma metodológico	38
Figura 7 - Página inicial do site Flutterflow.....	41
Figura 8 - Funcionalidades do FlutterFlow	42
Figura 9 - Página inicial do site Supabase.....	43
Figura 10 - Fluxo de uso do aplicativo	44
Figura 11 - Arquitetura de aplicação.....	45
Figura 12 - Preenchimento do campo com a fórmula do NPR	47
Figura 13 - Código para a condição da coluna “Status” no banco de dados	48
Figura 14 - Esquema da lógica de relacionamento entre as tabelas no Supabase ...	49
Figura 15 - Pasta de mídias no banco de dados	50
Figura 16 - Integração entre o banco de dados e FlutterFlow	51
Figura 17 - Área de desenvolvimento da plataforma FlutterFlow.....	52
Figura 18 - Estrutura da tela Criar conta/ Entrar	53
Figura 19 - Animação atribuída à tela	54
Figura 20 - Sequência de ações para o botão Criar conta.....	55
Figura 21 - Sequência de ações para o botão Entrar	55
Figura 22 - Estrutura do componente de entrada dos dados da edificação	56
Figura 23 - Estrutura da componente de editar ou deletar os dados da edificação ...	57
Figura 24 - Sequência de ações para o botão Salvar do componente dos dados da edificação	58
Figura 25 - Ação para o botão Fechar do componente dos dados da edificação.....	58
Figura 26 - Sequência de ações para o botão Editar do componente dos dados da edificação	59
Figura 27 - Filtro para a busca e exibição dos dados.....	60
Figura 28 - Sequência de ações para o botão Deletar do componente dados da edificação	61
Figura 29 - Estrutura do componente de entrada de dados da manifestação patológica	

.....	62
Figura 30 - Sequência de ações para o upload da foto	63
Figura 31 - Estrutura do componente de editar ou deletar os dados da manifestação patológica	64
Figura 32 - Estrutura do componente Meu perfil.....	65
Figura 33 - Atribuição da tabela "Usuario" ao container do componente Meu perfil ..	65
Figura 34 - Sequência de ações atribuídas ao botão Sair do componente Meu perfil	66
Figura 35 - Estrutura da tela Início	67
Figura 36 - Filtro dos dados para o usuário logado	68
Figura 37 - Filtro dos dados da edificação para usuário logado.....	68
Figura 38 - Lista dos dados da edificação declarada no Generate Dynamic Children	69
Figura 39 - Ação atribuída ao botão Menu.....	70
Figura 40 - Ação atribuída ao botão Conta do usuário presente no Menu	70
Figura 41 - Ação atribuída ao botão Ajuda presente no Menu.....	70
Figura 42 - Ação atribuída ao botão Editar	70
Figura 43 - Ação atribuída ao botão Adicionar	71
Figura 44 - Estrutura da tela Ajuda.....	71
Figura 45 - Ação do botão de acesso à tela Matriz FMEA	72
Figura 46 - Estrutura da tela Matriz FMEA.....	73
Figura 47 - Parâmetro para tela Matriz FMEA.....	74
Figura 48 - Filtro na tabela de manifestações patológicas conforme o ID da edificação	74
Figura 49 - Vínculo e filtro aplicado ao container "InfoMatrizFMEA"	75
Figura 50 – "DataTable" vinculada às linhas da tabela do banco de dados	75
Figura 51 - Condicionante de exibição dos valores de NPR na tabela "DataTable" ..	76
Figura 52 - Botão de gerar PDF com vínculo e filtro	77
Figura 53 - Definição dos argumentos e dependências.....	77
Figura 54 - Bloco básico fornecido pelo Boilerplate Action Code.....	78
Figura 55 - Bibliotecas adicionais.....	78
Figura 56 - Código para verificação das informações.....	79
Figura 57 - Código para organização do layout e cabeçalho do PDF.....	80
Figura 58 - Código para a organização do layout da tabela	81

Figura 59 - Continuação do código para a organização do layout da tabela	82
Figura 60 - Continuação do código para a organização do layout da tabela	83
Figura 61 - Continuação do código para a organização do layout da tabela	84
Figura 62 - Código para busca e organização das informações	85
Figura 63 - Telas Entrar e Criar conta	86
Figura 64 - Sequência de telas para o registro da edificação	87
Figura 65 - Sequência de telas para edição dos dados.....	88
Figura 66 - Sequência de telas para cadastrar as manifestações patológicas	89
Figura 67 - Sequência de telas para edição dos dados.....	90
Figura 68 - Exemplo do documento PDF	91
Figura 69 - Sequência de telas para sair da conta do aplicativo	92
Figura 70 - Tela de Ajuda com as informações dos índices da FMEA	93
Figura 71 - Criação da conta no aplicativo.....	96
Figura 72 - Sequência de telas para o cadastro da edificação.....	97
Figura 73 - Cadastro das manifestações patológicas	98
Figura 74 - Visualização geral da tabela no aplicativo	99
Figura 75 – Visualização das informações na tela do celular	100
Figura 76 - Documento PDF gerado	101
Figura 77 - Continuação do documento PDF gerado.....	102
Figura 78 - Informações exibidas no banco de dados	103

LISTA DE TABELAS

Tabela 1 - Classificação dos graus de risco.....	22
Tabela 2 - Origens da umidade.....	24
Tabela 3 - Classificação da anomalia conforme o tamanho da abertura	24
Tabela 4 - Índices da FMEA para Severidade (S).....	33
Tabela 5 - Índices da FMEA para Ocorrência (O)	33
Tabela 6 - Índices da FMEA para Detecção (D).....	34
Tabela 7 - Índices da FMEA para NPR.....	34
Tabela 8 - Atributos para as tabelas criadas no Supabase	46
Tabela 9 - Tabela de descrição dos recursos utilizados para a criação das telas.....	52
Tabela 10 - Aplicação da metodologia FMEA.....	95
Tabela 11 - Resultado obtido pela hierarquização em ordem decrescente	96

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
API	<i>Application Programming Interface</i>
CRUD	<i>Create, Read, Update, Delete</i>
FMEA	<i>Failure Modes and Effects Analysis</i>
GUT	Gravidade, Urgência e Tendência
HTTP	<i>Hypertext Transfer Protocol</i>
IBAPE	Instituto Brasileiro de Avaliações e Perícias de Engenharia
IFPI	Instituto Federal de Educação, Ciência e Tecnologia do Piauí
iOS	<i>iPhone Operating System</i>
MVP	<i>Minimum Viable Product</i>
NASA	<i>National Aeronautics and Space Administration</i>
NBR	Norma Brasileira Regulamentadora
NPR	Número de Prioridade de Risco
PDF	<i>Portable Document Format</i>
SQL	<i>Structured Query Language</i>
WEB	<i>World Wide Web</i>

SUMÁRIO

1 INTRODUÇÃO	14
1.1 CONTEXTUALIZAÇÃO	14
1.2 JUSTIFICATIVA.....	16
1.3 OBJETIVOS DA PESQUISA	17
1.3.1 Objetivo principal.....	17
1.3.2 Objetivos específicos.....	17
1.4 DELIMITAÇÕES	17
1.5 DELINEAMENTO	18
2 DESENVOLVIMENTO TEÓRICO	19
2.1 INSPEÇÃO PREDIAL	19
2.1.2 Manifestações patológicas.....	23
2.1.3 Planejamento da manutenção	26
2.2 GERENCIAMENTO DE RISCO	27
2.3 FMEA.....	28
2.3.1 Definição	28
2.3.2 Aplicação.....	30
2.3.3 Aplicação da FMEA na Engenharia civil	35
2.4 APLICATIVOS MÓVEIS E PLATAFORMAS <i>LOW-CODE</i>	36
3 METODOLOGIA DA PESQUISA.....	38
3.1 IDENTIFICAÇÃO DO PROBLEMA E DEFINIÇÃO DOS OBJETIVOS.....	38
3.2 REQUISITOS PARA A ESCOLHA DA PLATAFORMA DE DESENVOLVIMENTO	39
3.3 SELEÇÃO DA PLATAFORMA E SUAS INTEGRAÇÕES	40
3.3.1 FlutterFlow	40
3.3.2 Supabase.....	42
3.4 MODELAGEM DO APLICATIVO	43

3.4.1 Criação do fluxo de uso do aplicativo.....	43
3.4.2 Arquitetura do aplicativo.....	45
3.4.3 Modelagem dos dados	45
3.4.4 Desenvolvimento do <i>frontend</i>	51
3.4.4.1 Tela Criar conta/Entrar.....	53
3.4.4.2 Formulários	56
3.4.4.3 Tela Início	67
3.4.4.4 Tela Matriz FMEA	72
4 RESULTADOS.....	86
4.1 INTERFACES GRÁFICAS DO APLICATIVO.....	86
4.2 TESTE DE FUNCIONALIDADE DO APLICATIVO.....	94
5 DISCUSSÃO	104
6 CONSIDERAÇÕES FINAIS	106
7 RECOMENDAÇÕES.....	107
REFERÊNCIAS BIBLIOGRÁFICAS	108

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

A Construção Civil está sujeita a evolução das sociedades e de suas necessidades construtivas. Assim, a história da Engenharia Civil foi moldada pela busca por aprimoramento e inovação, tornando as edificações mais complexas devido ao constante desenvolvimento do conhecimento técnico-científico e à adequação às circunstâncias socioeconômicas de cada época. Diante disso, as obras atingiram mais velocidade de execução e, conseqüentemente, menor rigor de controle em relação aos serviços, materiais e planejamento, o que provocou a diminuição da qualidade das estruturas (Brito, 2017). Isso, por sua vez, resultou em frequentes aparecimentos de falhas nas edificações.

As causas para o aparecimento de manifestações patológicas nas estruturas não estão relacionadas apenas a fatores intrínsecos, que se referem a falhas desde a concepção até a execução, mas também a fatores extrínsecos à estrutura, como a influência das condições do meio ambiente e uso, às quais a edificação está exposta ao longo do tempo. Anomalias nas estruturas, como fissuras, infiltrações e corrosão de armaduras, indicam que a integridade da edificação está comprometida e que intervenções são necessárias. Muitas vezes, esses sinais demoram para serem identificados, o que pode levar ao comprometimento da segurança dos usuários da edificação. Assim, é importante detectar a presença das manifestações patológicas em suas fases iniciais, por meio de medidas preventivas, como a realização de inspeções regulares para a identificação dos problemas e compreensão das suas causas, possibilitando a definição de tratamentos eficazes e assegurando a durabilidade, segurança e conforto aos usuários da edificação.

A necessidade de planejamento de manutenção com rotinas de inspeção, permite a identificação dos estágios iniciais das falhas, evitando que se tornem mais graves e difíceis de serem resolvidas. Nesse sentido, a incorporação de metodologias que permitem analisar e apontar as causas, as ocorrências e evitar desvios de padrões, está sendo amplamente adotada na área da construção civil, com o objetivo de otimizar a gestão da qualidade e mitigação de riscos.

A Norma de Inspeção Predial Nacional (IBAPE, 2012) indica em um de seus itens, como parte do método do processo de inspeção predial, a definição da ordem

de prioridade das anomalias e falhas, caracterizando-se como uma etapa crucial para a gestão de problemas encontrados em uma edificação. Esse processo, por meio de metodologias de gestão de riscos, orienta a elaboração de estratégias mais eficazes e eficientes, possibilitando a estruturação da priorização de ações corretivas e tornando os planos de manutenção das edificações mais assertivos. Isso resulta em uma avaliação sistemática e criteriosa acerca das anomalias apontadas durante a inspeção.

As metodologias mais comuns, apontadas pela Norma de Inspeção Predial são: a matriz GUT (Gravidade, Urgência e Tendência) e a FMEA (*Failure Mode and Effects Analysis* - Análises de Modos de Falhas e Efeitos). Em ambos os métodos, obtém-se como resultado a ordem de priorização dos riscos. No entanto, de acordo com os estudos comparativos de Leite (2019), a matriz GUT é um processo mais simplificado, voltado exclusivamente para a hierarquização das falhas. Enquanto isso, a FMEA trata-se de uma abordagem mais detalhada, na qual se analisa o histórico da edificação, identificam-se as causas e propõem-se medidas corretivas.

A FMEA é uma opção de ferramenta eficaz e preventiva, que pode ser utilizada em diversos segmentos, como em atividades de inspeções prediais para mapear um caminho crítico para a elaboração de planos de manutenção, através da identificação das causas e definição de soluções dos problemas de forma estruturada e organizada. Em um panorama geral, essas metodologias dependendo de fatores, como o escopo e a complexidade do problema, demandam mais tempo para quantificação e hierarquização dos dados coletados.

O avanço das tecnologias nos últimos anos tornou-se um facilitador para o desenvolvimento de *softwares* que automatizam tarefas dispendiosas. As plataformas que utilizam *low-code* ou baixo código para o desenvolvimento de aplicativos surgiram como resposta à necessidade de reduzir a complexidade e tornar desenvolvimento de *softwares* mais rápido e eficiente (Silva, 2023). O *FlutterFlow* é um exemplo desse tipo de ferramenta, que permite por meio de um ambiente intuitivo e com recursos integrados, a criação de aplicativos móveis e *web* sem a exigência de codificação elaborada. A utilização dessas tecnologias acelera o processo de criação de MVPs (*Minimum Viable Products*) ou protótipos, cujo objetivo é testar e validar uma ideia que resolva lacunas de um problema específico no âmbito aplicado.

Em face ao exposto, o presente estudo visou utilizar os fundamentos da metodologia FMEA no setor da construção civil, através do desenvolvimento de um

aplicativo *low-code*. A premissa é criar uma ferramenta a ser utilizada em inspeções prediais para auxiliar o processo de mapeamento e categorização das manifestações patológicas de uma edificação, conforme sua severidade, e gerar relatórios em PDF para documentação e monitoramento da eficácia das ações corretivas. Assim, através dessa aplicação móvel, pretende-se contribuir tanto com uma solução prática e funcional para os profissionais da área quanto com o conhecimento acerca da importância da gestão de riscos associado à cultura de inspeções em edifícios.

1.2 JUSTIFICATIVA

A investigação das manifestações patológicas através de seu levantamento por meio de inspeções e a implementação de um programa de manutenção são pontos fundamentais para que se garanta a qualidade e durabilidade de uma edificação. Para Ortiz (2015), a Inspeção Predial deve ser tratada como uma vistoria destinada a avaliar o estado de conformidade de uma edificação mediante a aspectos como desempenho e segurança. Ainda consoante ao autor, as inspeções fornecem aos síndicos um direcionamento dos serviços a serem realizados, gerenciando melhor os recursos.

De acordo com Tutikian e Pacheco (2013), a realização de inspeções periódicas subsidiam as atividades de manutenção, pois possibilita a identificação das falhas e, por meio da priorização de ações corretivas, orientam a decisão sobre quais áreas demandam intervenção imediata. Dessa forma, a aplicação de metodologias de gerenciamento de riscos, como a FMEA, torna-se uma opção estratégica, baseada em critérios objetivos, oferecendo uma abordagem organizada para a análise das manifestações patológicas e proporcionando maior precisão na tomada de decisões.

A integração da tecnologia com a engenharia permite que a utilização dessa metodologia se torne ágil e promissora. A incorporação de um suporte tecnológico possibilita automatização de tarefas repetitivas e manuais, o que facilita o gerenciamento dos dados coletados e o monitoramento das intervenções já implementadas. Para isso, as plataformas de desenvolvimento em *low-code* são soluções modernas para a elaboração de mecanismos virtuais.

Diante disso, o estudo tem como objetivo alinhar conhecimentos de gestão de riscos e tecnologia através do desenvolvimento de um aplicativo móvel como suporte para auxiliar os profissionais em inspeções prediais, promovendo *insights* valiosos

para o planejamento de manutenções e uma melhor organização das informações. Além disso, busca-se incentivar a comunidade científica de acadêmicos para a realização de pesquisas sobre novas abordagens de gestão de riscos integradas com as tecnologias.

1.3 OBJETIVOS DA PESQUISA

1.3.1 Objetivo principal

Desenvolver um aplicativo *low-code*, utilizando os fundamentos do método de gestão FMEA para auxiliar no processo de hierarquização dos riscos das manifestações patológicas e agilizar a coordenação das atividades de inspeções.

1.3.2 Objetivos específicos

- Desenvolver, por meio da plataforma *FlutterFlow*, um aplicativo móvel com uma interface intuitiva e funcional baseado na metodologia FMEA.
- Desenvolver funcionalidades dentro da ferramenta para facilitar o processo de organização das informações, bem como possibilitar a geração de relatórios, baseado nos dados coletados durante as inspeções.
- Testar a funcionalidade do aplicativo por meio de um estudo de caso, demonstrando a eficácia da sua utilização em atividades de inspeções prediais.
- Cooperar com a formação de cabedal teórico e prático acerca do tema e impulsionar pesquisas subsequentes.

1.4 DELIMITAÇÕES

A presente pesquisa está delimitada ao desenvolvimento de um aplicativo móvel em fase inicial, focando na concepção e prototipagem da ideia, com aplicação no estudo de caso realizado por Regos (2022) para o teste de funcionalidade. Assim, busca-se alimentar o aplicativo com os dados coletados pela autora na edificação estudada e gerar um documento PDF com a lista de problemas identificados e hierarquizados, utilizando a metodologia FMEA.

1.5 DELINEAMENTO

O presente trabalho de conclusão está dividido em 7 (sete) capítulos, elaborados a partir de uma revisão de literatura, garantindo, assim, que as conclusões tenham resultados bem fundamentados, proporcionando uma melhor compreensão do estudo. O trabalho, portanto, está seccionado como descrito a seguir:

Capítulo 1 - O atual capítulo apresenta a introdução ao tema, justificativa, bem como sua delimitação e os objetivos principais e específicos que nortearão toda a pesquisa.

Capítulo 2 - Engloba uma revisão da literatura sobre os aspectos gerais relacionados à inspeção predial em edificações, às principais manifestações patológicas das construções e ao planejamento de manutenção. Discute, também sobre o gerenciamento de risco, a metodologia FMEA, aplicativos móveis e plataformas *low-code*.

Capítulo 3 - Este capítulo apresenta os métodos, descrevendo a plataforma virtual, integrações necessárias e o processo de desenvolvimento do aplicativo.

Capítulo 4 - Nesta seção é apresentado os resultados com a apresentação final das interfaces gráficas do aplicativo e o teste de funcionalidade utilizando o estudo de caso de Regos (2022).

Capítulo 5 - É apresentado a discussão acerca do processo de desenvolvimento do aplicativo e dos resultados obtidos com teste de funcionalidade.

Capítulo 6 - Neste capítulo são feitas as considerações finais do trabalho com a reflexão dos resultados alcançados e as contribuições do estudo.

Capítulo 7 - É apresentado as recomendações para o aplicativo com sugestões de ampliação dos seus recursos.

2 DESENVOLVIMENTO TEÓRICO

O conteúdo apresentado neste capítulo fornecerá uma base essencial para discernimento do assunto abordado no presente estudo, expondo conceitos-chave e fundamentos relacionados ao tema. Assim sendo, o tópico abordará os seguintes temas: Inspeção Predial, Gerenciamento de risco, FMEA e Aplicativos móveis e plataformas *low-code*.

2.1 INSPEÇÃO PREDIAL

A inspeção predial refere-se à avaliação técnica de suma importância para que se tenha a garantia da funcionalidade da edificação. É, portanto, a atividade que verifica as condições de segurança, funcionamento e conservação do edifício. Segundo Gomide *et al.* (2006), a inspeção predial é uma ferramenta que obtém informações das conformidades e não conformidades, concentrada na melhoria da qualidade e durabilidade das estruturas. Ou seja, essa atividade avalia o cumprimento das exigências estabelecidas em normas que englobam critérios para assegurar o desempenho da edificação.

A NBR 16747 (ABNT, 2020), define como um processo de avaliação sensorial, no qual se visa auxiliar na gestão da edificação e quando realizada periodicamente, facilita a mitigação dos riscos correlacionados ao desempenho da estrutura. Tendo em vista isso, a inspeção predial analisa o estado da edificação para identificar a necessidade da realização de reparos ou ações corretivas e preventivas, com o objetivo de assegurar sua longevidade e evitar o aparecimento de problemas que comprometam a integridade dos sistemas ou da estrutura.

A garantia de qualidade e segurança de uma estrutura é norteadada pela realização de vistorias preventivas executadas por engenheiros e arquitetos legalmente habilitados. A solicitação da realização da inspeção predial cabe ao responsável do imóvel ou síndico. A fiscalização de um edifício é fundamental para evitar o diagnóstico tardio da falha, o que implica em um maior direcionamento de recursos financeiros para sua resolução, devido à necessidade de reparos mais extensos que podem levar à substituição total de um componente.

A inspeção predial é um trabalho investigativo que considera em seu processo de avaliação, as técnicas construtivas e as condições de manutenção e uso de uma

edificação. A partir disso, lista-se todos os sistemas e subsistemas que serão analisados para lucidar a elaboração do laudo técnico que conterá a avaliação de tudo o que foi observado durante a vistoria e as medidas de manutenção que devem ser realizadas com urgência. Por meio da priorização dos problemas identificados, o síndico ou responsável pelo imóvel organiza e define planos financeiros para execução das ações corretivas, através das informações fornecidas no laudo da inspeção predial.

A complexidade da análise da edificação delimitará se abordagem da avaliação será sensorial ou mais técnica e profunda com a utilização de equipamentos especializados. Ademais, o desenvolvimento desse processo deve ser implementado de maneira planejada e conforme as características da edificação. A NBR 16747 (ABNT, 2020) apresenta as seguintes etapas da inspeção predial:

- a) levantamento de dados e documentação;
- b) análise dos dados e documentação solicitados e disponibilizados;
- c) anamnese para a identificação de características construtivas da edificação, como idade, histórico de manutenção, intervenções, reformas e alterações de uso ocorridas;
- d) vistoria da edificação de forma sistêmica, considerando a complexidade das instalações existentes;
- e) classificação das irregularidades constatadas;
- f) recomendação das ações necessárias para restaurar ou preservar o desempenho dos sistemas, subsistemas e elementos construtivos da edificação afetados por falhas de uso operação ou manutenção, anomalias ou manifestações patológicas constatadas e/ou não conformidade com a documentação analisada (considerando, para tanto, o entendimento dos mecanismos de deterioração atuantes e as possíveis causas das falhas, anomalias e manifestações patológicas);
- g) organização das prioridades, em patamares de urgência, tendo em conta as recomendações apresentadas pelo inspetor predial;
- h) avaliação da manutenção, conforme a ABNT NBR 5674;
- i) avaliação do uso;
- j) redação e emissão do laudo técnico de inspeção.

De acordo com os passos apresentados na norma, é possível perceber que a estruturação do processo de inspeção predial analisa a edificação em diversos âmbitos, fornecendo uma visão detalhada da edificação e sintetizada em um laudo.

Esse documento atesta o cumprimento da NBR, reúne e organiza todas as informações coletadas na vistoria. Além disso, atua como histórico para consultas posteriores, fornecendo uma base fundamental para o acompanhamento da condição da edificação ao longo do tempo e para gestão da manutenção.

Em resumo, Bigolin *et al.* (2014), destaca o desenvolvimento da inspeção predial em etapas apresentadas na Figura 1:

Figura 1 - Etapas do desenvolvimento da atividade de Inspeção Predial



Fonte: Bigolin *et al.* (2014).

A inspeção predial é classificada por Gomide *et al.* (2009), de modo geral em:

- Inspeção “in loco”: atividade realizada no próprio local.
- Inspeção indireta: processo em que ocorre por meio de registros, seja fotográfico ou de laudos, relacionados ao fato.
- Inspeção sumária: processo em que se obtém uma análise sucinta de um fato ou condição, usualmente, sem registros fotográficos.
- Inspeção ilustrada: atividade realizada “in loco” com fotografias para ilustração seguida de fundamentação.
- Inspeção detalhada: realizada “in loco” acompanhada de registros fotográficos e descritivos para ilustração e posterior fundamentação detalhada.

A Norma de Inspeção Predial Nacional (IBAPE, 2012), aponta níveis de classificação da inspeção mediante a sua dificuldade. O nível 1 (um) é o tipo de inspeção de baixa complexidade em que os resultados são baseados na observação do profissional técnico. Já o nível 2 (dois), se constitui na vistoria realizada em

conjunto com outros profissionais em edificações de complexidade técnica média. O nível 3 (três) é quando envolve edificações com alta complexidade técnica e sofisticação, sendo assim, a inspeção é conduzida por diversos profissionais com a utilização de ferramentas de gestão, como *softwares*.

As irregularidades de um sistema, delineadas por anomalias e falhas, são citadas com conceitos diferentes na NBR 16747 (ABNT, 2020) e, quando determinadas, direcionam os planos de reparo. As anomalias trata-se da perda de desempenho do elemento por causas endógenas, exógenas e funcionais. Enquanto as falhas são decorrentes do uso, operação e manutenção. Quanto a análise do grau de risco das irregularidades, é considerada pela Norma de Inspeção Predial em três níveis, sendo crítico, médio e mínimo, como apresentado na Tabela 1.

Tabela 1 - Classificação dos graus de risco

Grau de Risco	Descrição do risco
Crítico	Risco de provocar danos contra a saúde e segurança das pessoas e do meio ambiente; perda excessiva de desempenho e funcionalidade causando possíveis paralisações; aumento excessivo de custo de manutenção e recuperação; comprometimento sensível de vida útil.
Médio	Risco de provocar a perda parcial de desempenho e funcionalidade da edificação sem prejuízo à operação direta de sistemas, e deterioração precoce.
Mínimo	Risco de causar pequenos prejuízos à estética ou atividade programável e planejada, sem incidência ou sem a probabilidade de ocorrência dos riscos críticos e regulares, além de baixo ou nenhum comprometimento do valor imobiliário.

Fonte: IBAPE (2012).

Tendo em vista as proporções da extensão que um dano pode alcançar, é necessário que haja o reparo nos sistemas estruturais afetados para evitar o deterioramento progressivo e geral da estrutura. A par disso, o processo da atividade de inspeção predial possibilita uma compreensão abrangente em torno de toda a edificação. O conhecimento sobre as manifestações patológicas presentes em uma edificação junto com discernimento dos riscos e gravidade, permite analisar os perigos nas quais os usuários e o patrimônio estão expostos, além de planejar e gerenciar melhor as intervenções.

2.1.2 Manifestações patológicas

As manifestações patológicas tratam-se das falhas visíveis ou observáveis nas estruturas, sendo assim, o objeto principal de estudo da Patologia das construções. Essas falhas são indicadores que apontam, de maneira geral, a presença de problemas na edificação que são originados por diversos fatores, permitindo que seja classificado conforme o tipo, causa e a ocorrência em determinada etapa do processo construtivo (Oliveira, 2013). A preocupação no setor da construção civil com as manifestações patológicas e compreensão das suas causas reflete a necessidade de garantir a durabilidade das edificações. As falhas mais comuns encontradas em edificações, segundo Santos *et al.* (2017), são: a umidade, corrosão de armaduras, degradação do concreto, eflorescência, fissuras, trincas e rachaduras. Antunes (2010), por sua vez, ainda pontua como falhas frequentes, o descolamento de revestimento, principalmente em fachadas.

a) Umidade

A água é um dos principais fatores que contribui para a deterioração dos materiais de construção e para o surgimento de defeitos nas edificações, consoante a NBR 15575 – Parte 1 (ABNT, 2024). A formação de manchas em tetos e paredes, geralmente ocorre devido ao acúmulo de umidade, resultado do impedimento da água de se dissipar ou escoar corretamente para outro local. Dessa forma, a umidade pode se manifestar em diferentes formas, como mofo, bolor e eflorescência, e simultaneamente em diversas partes de uma edificação, o que compromete a estética e em casos mais graves pode desencadear problemas estruturais, motivo pelo qual é importante buscar a causa raiz e os tratamentos necessários. A Tabela 2, aponta as principais origens da umidade, de acordo com Machado (2019).

Tabela 2 - Origens da umidade

Origens da umidade	Ocorrências
De construção	Nas confecções do concreto e argamassas, pinturas e entre outros.
De condensação	Coberturas, lajes de terraço e paredes
De capilaridade	Através de lençol freático e terra
De infiltração	Por meio de trincas, fissuras, redes hidráulicas e impermeabilização

Fonte: Machado (2019).

b) Fissuras, trincas e rachaduras

Fissuras, trincas e rachaduras são problemas comuns de serem detectados nas edificações e, embora muitas vezes sejam confundidas, cada uma tem características e gravidades diferentes. Thomaz (1989) menciona como causas das ocorrências dessas manifestações patológicas em questão: o recalque de fundações, a retração de produtos à base de cimento, a deformabilidade excessiva das estruturas, a atuação de sobrecargas, as movimentações térmicas e higroscópicas e as alterações químicas dos materiais. Dentro desse contexto, é possível, dependendo da causa, classificar a anomalia de acordo com o tamanho da sua abertura. Oliveira (2012) classifica em cinco tipos: fissura, trinca, rachadura, fenda ou brecha, como mostrado na Tabela 3.

Tabela 3 - Classificação da anomalia conforme o tamanho da abertura

Anomalia	Abertura
Fissura	$\leq 0,5 \text{ mm}$
Trinca	$0,5 \leq e \leq 1,5 \text{ mm}$
Rachadura	$1,5 \leq e \leq 5,0 \text{ mm}$
Fenda	$5,0 \leq e \leq 10,0 \text{ mm}$
Brecha	$\geq 10,0 \text{ mm}$

Fonte: Adaptado Oliveira (2012).

c) Descolamento de revestimento cerâmicos em paredes e pisos

O descolamento de revestimentos cerâmicos acontece quando as peças de cerâmica de sistemas verticais ou horizontais se desprendem da base sobre a qual

foram fixadas. Uma das razões para essas falhas nos revestimentos é a movimentação do material devido às variações de temperatura, o que é recorrente em fachadas de prédios, já que estão expostas às intempéries adversas. A ausência de juntas de movimentação e de dessolidarização associados a qualidade dos materiais e ao assentamento inadequado também se configuram como causas para o aparecimento dessas manifestações patológicas.

d) Descascamento e empolamento da pintura

A aplicação de pintura sobre as superfícies tem como objetivo a proteção contra desgastes externos, a durabilidade das edificações e o acabamento. Em contrapartida, a tinta perde sua aderência ao longo do tempo e começa a se desprender em camadas, comprometendo tanto a aparência quanto sua função de conservação (Alves, 2010). Entre as causas para o aparecimento dessas falhas estão a preparação inadequada da superfície de reboco, a existência de camadas antigas de tinta, aplicação em bases úmidas, bem como a formação de eflorescência e bolores devido à umidade (Peres, 2001).

e) Corrosão de armaduras

Segundo Santos (2017), as armaduras sofrem corrosão gradual, já que o processo se inicia na parte externa do concreto e avança para seu interior, comprometendo a integridade estrutural. A seção de aço resistente se transforma em ferrugem, um subproduto do avanço da corrosão, que leva a redução da seção transversal do aço, perda de aderência e a uma expansão volumétrica. Esse processo gera uma pressão interna que excede a capacidade do concreto de absorver essa mudança, provocando o aparecimento de fissuras e a desagregação do material. A corrosão é potencializada quando a estrutura está exposta à ambientes agressivos, como aqueles com alta umidade ou aqueles ricos em cloretos (em áreas costeiras ou onde sais são usados para degelo). A NBR 6118 (ABNT, 2023), coloca a carbonatação do concreto e a presença de íons de cloreto como fenômenos principais que contribuem para a despassivação das armaduras, levando-as ao processo corrosivo.

2.1.3 Planejamento da manutenção

A durabilidade e desempenho são interdependentes e devem ser analisados em todas as etapas do ciclo de vida da edificação, podendo ser tratada desde o projeto até a manutenção, que se prolonga ao longo dos anos. Em conformidade com a ISO 13823 (2008), a durabilidade se refere a capacidade de uma estrutura ou mesmo de seus componentes, aliada com manutenção planejada, de satisfazer requisitos de desempenho previstos em projeto por determinado período sob efeito de ações ambientais ou por influência do processo de envelhecimento natural.

A prática da manutenção predial é um fator de grande impacto para manter o edifício seguro, funcional e valorizado com o passar do tempo. Essa atividade inclui tanto contestar a existência de problemas através da inspeção predial, quanto a correção e a antecipação de problemas futuros, o que assegura que a estrutura atenda às normas legais, preservando a integridade e a segurança dos seus ocupantes. Em suma, destinar recursos para a elaboração de um plano de manutenção com monitoramento contínuo significa evitar despesas elevadas com reparos emergenciais, prevenir acidentes e proteger o patrimônio.

A partir do gráfico (Figura 2), é possível notar que a falta de manutenção é um dos fatores preponderantes para o aparecimento de falhas no sistema construtivo, como afirma os resultados apontados pelo IBAPE/SP (2015):

Figura 2 - Distribuição da incidência dos acidentes prediais por origem



Fonte: IBAPE/SP (2015).

Sendo assim, a inspeção predial desempenha um papel crucial quando realizada com periodicidade, auxiliando na gestão predial e servindo como ferramenta para que o síndico tenha o conhecimento do estado de conservação do edifício, e assim administrar os recursos financeiros e priorizar intervenções mais urgentes. Com isso, o desenvolvimento de rotinas de inspeções prediais em uma determinada edificação, fornece informações analíticas, o que permite a criação de um plano de manutenção preventiva, colaborando para evitar despesas altas com reparos não programados. Além disso, é um aspecto relevante para a valorização do patrimônio, uma vez que, certifica-se que o imóvel continue em condições adequadas, agregando valor no mercado e proporcionando melhor qualidade de vida para os habitantes.

2.2 GERENCIAMENTO DE RISCO

O gerenciamento de risco no processo de inspeção predial é fundamental para a identificação de falhas potenciais que possam comprometer a integridade, segurança e desempenho de um sistema. Schneider (2014) afirma que, desde o projeto até a sua finalização, o foco no gerenciamento de riscos é o ponto principal para prever e minimizar problemas que podem afetar o objetivo do projeto. Aplicado ao contexto de análise de manifestações patológicas, essa gestão permite a determinação de prioridades, o que auxilia o traçado de planos de manutenção predial com ações preventivas ou corretivas definidos mediante a urgência de cada falha

A NBR 16747 (ABNT, 2020), que trata da inspeção predial, estabelece a organização de prioridades em patamares de urgência, com base nas recomendações do responsável técnico pela inspeção. A hierarquização de acordo com a criticidade de um problema é aprimorada com a utilização de ferramentas de gestão da qualidade, o que otimiza os custos com manutenção corretiva e cria uma rotina de prevenção de falhas em uma edificação (Silva, 2015). Essas ferramentas avaliam de forma qualitativa e quantitativa, tornando a abordagem de gestão de riscos mais estratégica e estruturada, o que melhora a confiabilidade do sistema. Alguns exemplos de metodologias que ajudam na identificação de falhas e melhora de processos são:

- Diagrama de Ishikawa
- Metodologia FMEA (Análise de Modos de Falha e Efeitos)
- Análise SWOT (Forças, Fraquezas, Oportunidades e Ameaças)
- Matriz GUT (Gravidade, Urgência e Tendência)
- Método ABC (Classificação de Prioridades)
- Diagrama de Pareto

A utilização de ferramentas que possibilitam uma abordagem sistemática e, conseqüentemente, uma visualização gráfica e estatística, desempenha um papel determinante para se obter uma clareza no tratamento dos dados obtidos e conclusões mais fundamentadas sobre os riscos. Com isso, é possível analisar o impacto de cada risco em um sistema e assim mensurar e determinar soluções mais precisas para o problema identificado. Outro ponto é que a implementação dessas metodologias, por meio do monitoramento, permite a análise das ações corretivas adotadas, viabilizando revisões e aprimoramentos contínuos.

2.3 FMEA

2.3.1 Definição

A FMEA, sigla para *Failure Modes and Effects Analysis* (Análise de Modos de Falha e Efeitos Potenciais), é uma metodologia que tem sua origem nos Estados Unidos, desenvolvida em 1949 para a identificação de falhas nas operações militares (Sakurada, 2001). Assim, era possível classificá-las conforme os impactos nos sucessos das missões. Posteriormente, essa metodologia passou a ser utilizada no desenvolvimento de projetos como Apollo, pela NASA (*National Aeronautics and Space Administration*), e em indústrias automotivas.

Há diversos conceitos na literatura em torno da metodologia FMEA. Para Matos e Milan (2009), trata-se de um método analítico que busca detectar e eliminar problemas de forma sistemática e completa. Na visão de Mzougui e El Felsoufi (2019), é uma ferramenta poderosa e organizada que melhora a confiabilidade do sistema.

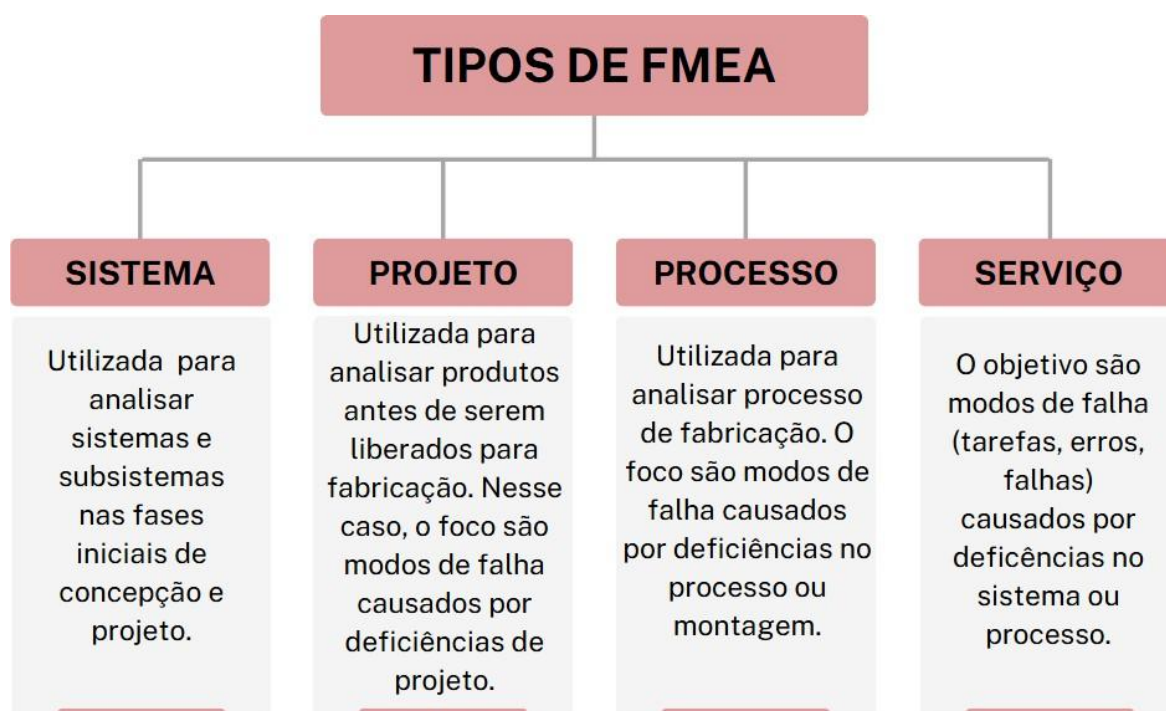
A definição dessa ferramenta de gestão consiste em um estudo que analisa possíveis modos de falhas dos sistemas, tal como os efeitos que são gerados em decorrência desses fatores. A FMEA, nesse sentido, é um procedimento que classifica

os modos de falha conforme a severidade dos seus efeitos (Toledo e Amaral, 2006). Assim, é possível priorizar e desenvolver estratégias para a eliminação ou mitigação de potenciais riscos.

Nesse caso, a metodologia é uma forma lógica de classificar cada modo de falha mediante a sua gravidade, levando em consideração sua severidade, a forma de ocorrência e detecção. Apesar da FMEA inicialmente ter sido desenvolvida com enfoque na previsão de falhas na fase de projeto de novos produtos, sua implementação em qualquer outra fase do sistema é viável, possibilitando também sua utilização após a manifestação do problema. Nesse sentido, é imprescindível a determinação do escopo de estudo para sua correta aplicação.

A FMEA é classificada em quatro tipos, consoante a Stamatis (2003): de sistema, de projeto, de processo e de serviço (Figura 3). Nessa classificação, cada um é aplicado a uma fase específica, o que possibilita às empresas definirem a mais adequada para sua situação, direcionando suas ações com mais eficácia às suas necessidades. Dessa forma, a versatilidade dessa ferramenta de gestão permite a adaptação da sua matriz de aplicação a diversas situações e setores.

Figura 3 - Classificação da FMEA



Fonte: Adaptado Stamatis (2003).

A aplicação dessa ferramenta, em um cenário global com intensa competitividade entre as organizações, é uma estratégia vital para que empresas de qualquer ramo permaneçam, ao menos, com um mínimo de lucros em suas atividades. Nesse sentido, sua abrangência para área da construção civil, principalmente, para gestão de riscos dentro das atividades de inspeções e manutenções prediais, busca prever e reduzir danos, visando à segurança das estruturas.

Moura (2000), destaca em síntese como objetivos da FMEA:

- Identificação e análise das falhas
- Avaliação dos riscos
- Desenvolvimento de ações e documentações do processo de análise

A identificação do risco é um processo contínuo de determinação das falhas que podem afetar o projeto, e é obtida por meio da coleta de informações, técnicas com diagramas e listas de verificações para a análise do impacto de cada falha. Já a avaliação é um processo em que os riscos são postos em ordem de priorização de acordo com seu impacto e ocorrência. Conjuntamente, o planejamento de ações envolve, conforme PMBOK (2013), o desenvolvimento de opções para a redução de ameaças de falhas potenciais, sendo implantadas abordagens de monitoramento para riscos residuais ou identificação de novos.

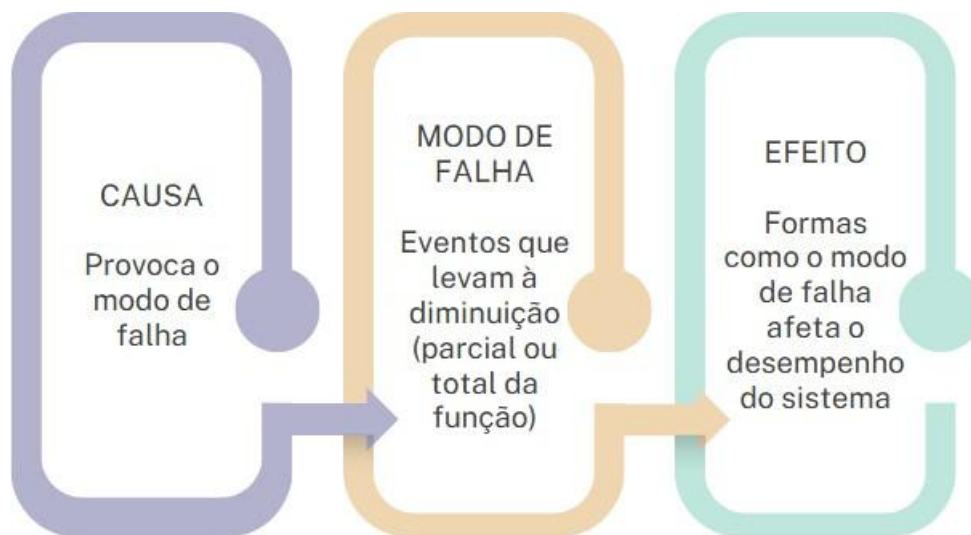
2.3.2 Aplicação

A FMEA consiste em duas fases, mediante a Leal *et al.* (2006). Na primeira, realiza-se a identificação das falhas durante as atividades de inspeção. Já na segunda fase, determina-se a criticidade em um ranking, com base em parâmetros já previamente estabelecidos. A literatura traz diversas etapas para a condução da metodologia FMEA dentro de uma área, todavia, esse fator é independente, já que o essencial é alcançar a detecção e diminuição da incidência de falhas.

Para um melhor usufruto dessa ferramenta de gestão é necessário que se tenha o conhecimento sobre modo de falha, efeito e causa. O modo de falha se configura na maneira ou forma como um sistema pode falhar, possuindo uma abordagem funcional (genérica) ou estrutural, que exige entendimento técnico. Em

contrapartida, o efeito de falha são as consequências do modo de falha. Já a causa é o fator que levou à falha do sistema.

Figura 4 - Relação Causa-Modo-Efeito

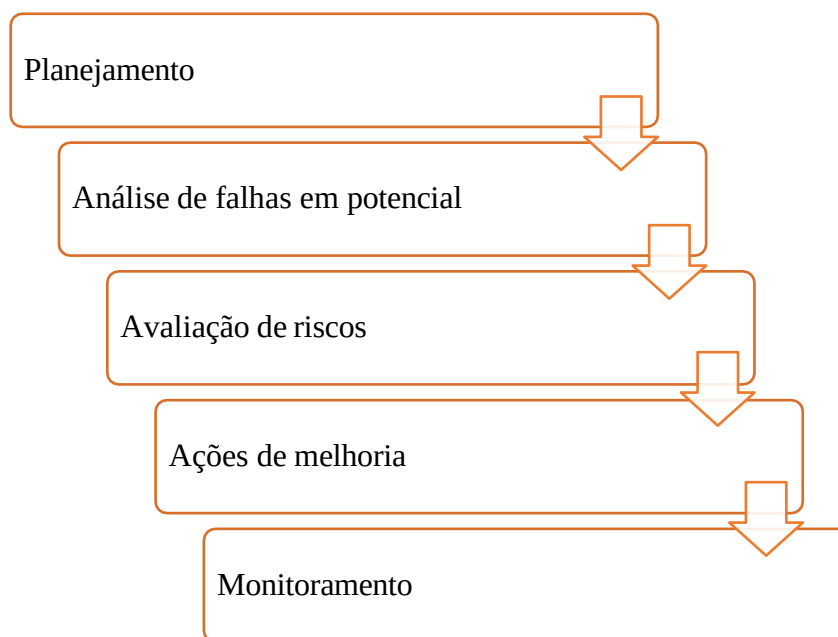


Fonte: Adaptado de Takayama (2008).

A FMEA, quando aplicada em determinado setor, funciona em cinco etapas, assim como é descrito por Toledo e Amaral (2006):

- Planejamento: envolve a organização de documentos e reuniões para o delineamento acerca da análise do produto ou serviço.
- Análise de falhas em potencial: onde são analisadas as causas e consequências de cada modo de falha, por meio de preenchimento de formulários.
- Avaliação de riscos: cada modo de falha é classificado seguindo critérios de severidade, ocorrência e detecção.
- Ações de melhoria: baseado nas análises, é estabelecido ações estratégicas para diminuir os riscos.
- Monitoramento ou continuidade: são revisões regulares para assegurar a identificação de novas falhas.

Figura 5 - Fluxograma de execução da FMEA



Fonte: Adaptado Toledo e Amaral (2006).

A implementação da metodologia FMEA consiste na análise de três fatores importantes para classificar e priorizar as falhas potenciais, através da obtenção do número de prioridade de risco (NPR). McCain (2006) aborda que o NPR engloba a avaliação da severidade (S), da probabilidade de ocorrência (O) e da probabilidade de detecção (D) da falha. O produto de tais índices possibilita a hierarquização dos riscos que demandam maior cautela e precisão na definição de estratégias de solução.

Kardec e Nascif (2012) destacam sete passos fundamentais para a constituição do formulário FMEA, com o objetivo de identificar a raiz do problema. A primeira etapa envolve reconhecer o componente que está com problemas, determinando seu modo de falha e as causas relacionadas. Em seguida, a segunda etapa consiste em analisar os efeitos dessa falha tanto para o próprio funcionamento do componente quanto para o comprometimento de terceiros, como o risco de ocorrência de acidentes.

O terceiro passo especifica o índice de severidade da falha, ou seja, o impacto caso não seja solucionada. A severidade está diretamente relacionada com o efeito, assim, quanto mais crítico o efeito é caracterizado, maior é seu índice de severidade. Nessa etapa, estabelecem-se pontuações que podem variar de 1 a 10 para a avaliação dos riscos em relação à sua gravidade, como apresentado na Tabela 4.

Tabela 4 - Índices da FMEA para Severidade (S)

Índice	Conceito
1	Falha de menor importância, quase não são percebidas
2-3	Provoca redução da performance da instalação. O cliente perceberá a falha, mas não ficará insatisfeito.
4-5-6	A instalação sofrerá uma degradação progressiva. O cliente perceberá a falha e ficará insatisfeito.
7-8	Mais de 50 a 70% das vezes não se consegue manter o desempenho da edificação. O cliente perceberá a falha e ficará muito insatisfeito
9-10	Os problemas são catastróficos e podem gerar danos a bens ou pessoas. O cliente ficará muito insatisfeito.

Fonte: Adaptado de Silva (2019).

Por conseguinte, realiza-se uma coleta de informações sobre a frequência, número estimado ou cumulativo da ocorrência do problema. Esse passo, baseia-se no modo de falha ou causa perante a sua possibilidade de manifestação. A precisão no levantamento dessa informação ajuda a direcionar as ações corretivas para o modo de falha mais crítico e, a partir daí, reduzir sua frequência. Na Tabela 5 está descrita cada índice conforme sua ocorrência.

Tabela 5 - Índices da FMEA para Ocorrência (O)

Índice	Ocorrência (O)
1	Muito remota
2-3	Muito pequena
4-6	Moderada; ocasional
7-8	Frequente
9-10	Inevitável, certamente ocorrerá a falha

Fonte: Adaptado de Silva (2019).

O quinto passo avalia a probabilidade de detecção, ou seja, a facilidade com a qual a falha pode ser identificada. Essa fase é fundamental para ter-se o conhecimento antes que o dano ocorra ou cause impactos indesejados para o sistema. Nesse caso, quanto maior o índice, menor será a possibilidade de identificação antes da falha se

manifestar.

Tabela 6 - Índices da FMEA para Detecção (D)

Índice	Detecção (D)
1	Muito alta probabilidade de detecção
2-3	Alta probabilidade de detecção
4-6	Moderada probabilidade de detecção
7-8	Pequena probabilidade de detecção
9	Muito pequena a probabilidade de detecção
10	Muito remota probabilidade de detecção

Fonte: Adaptado de Silva (2019).

No sexto passo, realiza-se, com base nos dados coletados, o cálculo do índice do Número de Prioridade de Risco (NPR), que é o produto dos parâmetros de Severidade (S), Ocorrência (O) e Detecção (D). Tal resultado facilita a classificação de prioridades e, em muitos casos, são utilizados valores pré-estabelecidos para a avaliação do risco, que pode variar conforme o responsável técnico. Nesse viés, quanto maior o valor encontrado do NPR, maior será a prioridade para tomada de ações corretivas. Para casos de empate, deve-se considerar outros parâmetros de análise, como, por exemplo, a extensão do dano e seu impacto na realização das atividades. Na Tabela 7, são apresentados os valores para NPR com a classificação dos riscos proposta por Sanches (2010).

Tabela 7 - Índices da FMEA para NPR

Avaliação	Pontuação de risco
Baixo	1 a 50
Médio	50 a 100
Alto	100 a 200
Muito alto	200 a 1000

Fonte: Adaptado de Sanches (2010).

O último passo, são as medidas específicas de manutenção ou estudos sugeridos para melhorar a detecção, reduzir ou erradicar a falha. Essas ações devem

ser definidas por um responsável técnico para a minimização dos índices que compõe o NPR, levando em consideração o custo, tempo e impacto.

Diante as diversas vantagens da FMEA, é importante ressaltar desafios de implementação. Conforme os autores Hawkins e Woollons (1998), um dos motivos para a dificuldade de aplicação dessa metodologia é o tempo gasto para sua elaboração, bem como no processamento das informações para a organização e definição de prioridades. Entretanto, com os avanços tecnológicos, tornou-se possível desenvolvê-la de maneira mais automatizada com auxílio da programação, permitindo maior facilidade e precisão na sua execução.

Essa metodologia deve ser aplicada de forma contínua, possibilitando a retenção do conhecimento adquirido e servindo como base para análises futuras, o que resulta na economia de tempo e aprimora a qualidade da avaliação (Barsalou, 2022). Diante disso, por ser uma documentação, ou seja, um registro de análises e decisões relacionadas às falhas, a FMEA se torna oportuna para o “amadurecimento” do conhecimento dos processos, o que diminui ou evita a probabilidade dos problemas vir a acontecer novamente.

2.3.3 Aplicação da FMEA na Engenharia civil

A FMEA apesar de ser uma ferramenta bastante difundida no setor industrial e automotivo, está ganhando cada vez mais espaço em outras áreas como a construção civil. Alguns trabalhos realizados que aplicaram a FMEA em diferentes etapas de projeto na área da Engenharia civil, adaptando o formulário de acordo com as necessidades de cada caso, confirmam as vantagens na otimização de processos e na melhora em aspectos como segurança e gastos com cenários não planejados.

Araújo (2019), em sua pesquisa, apresentou um estudo de caso de oito projetos arquitetônicos fornecidos por uma Empresa Júnior. O objetivo foi aplicar a FMEA para diagnosticar potenciais erros na fase de projeto e mostrar o impacto na execução em obra. Com isso, constatou-se que o método auxiliou na identificação dos problemas e orientou a adoção de medidas de controle. A autora obteve indicadores concretos, apontando que, quanto maior o volume de dados estudados, mais precisos serão os resultados.

Figueiredo *et al.* (2024), utilizaram em seus estudos a metodologia FMEA para a análise dos vícios construtivos e como auxílio na gestão de controle de serviços pós-

obra. Para isso, as autoras adaptaram a abordagem tradicional de mensuração dos riscos ao considerar custo, severidade e complexidade. As autoras concluíram que a FMEA é uma ferramenta proativa para a avaliação de riscos em um processo construtivo, elevação da confiabilidade, prevenção de falhas, além de contribuir para melhorias contínuas na assistência técnica.

Bittencourt (2021), por sua vez, através da realização de uma inspeção predial, aplicou a metodologia FMEA para priorizar as intervenções das manifestações patológicas identificadas nos sistemas prediais da Embrapa Mandioca e Fruticultura, no estado da Bahia. A conclusão do trabalho destaca que a hierarquização de prioridades contribui como parâmetro para o planejamento da manutenção das falhas, pontuando também a importância do conhecimento técnico para a determinação dos índices da FMEA e avaliação de impactos.

Diante dos estudos, verifica-se que a versatilidade dessa metodologia na construção civil é um ponto promissor para sua implementação em diferentes contextos, principalmente no âmbito da inspeção predial, em que se busca identificar falhas que podem colocar em risco o funcionamento dos sistemas prediais.

2.4 APLICATIVOS MÓVEIS E PLATAFORMAS *LOW-CODE*

Os aplicativos móveis são utilizados para desempenhar funções específicas como entreter ou facilitar atividades cotidianas dos usuários. Tratam-se de softwares desenvolvidos para uso em dispositivos móveis, como *smartphones* e *tablets*. A proliferação desses dispositivos somadas com as vantagens de dinamicidade, portabilidade e produtividade, tornou ampla a incorporação de aplicativos móveis nas diversas áreas do conhecimento, visando à automatização de processos.

Conforme a literatura, os aplicativos móveis são divididos em três tipos: aplicativos nativos, aplicativos *web* e aplicativos híbridos. Os aplicativos nativos referem-se aos softwares desenvolvidos para os sistemas operacionais Android ou iOS. Os aplicativos *web* são aqueles executados em uma interface de navegação, possibilitando a adequação para o tipo de dispositivo que está acessando o site (Silva e Prado, 2019). Já os do tipo híbrido, consistem na combinação dos aplicativos nativos e *web*, ou seja, são *websites* empacotados para funcionar como um aplicativo nativo (IBM, 2024).

O processo de desenvolvimento de aplicativos envolve etapas bem definidas e

planejadas para que o software atenda aos requisitos estabelecidos e às necessidades da demanda que pretende ser atendida. Sob esse viés, estudar as plataformas e linguagens de programação disponíveis no mercado é um ponto crucial para avaliar a opção que melhor se adequa à proposta que se busca desenvolver e implementar.

As plataformas de desenvolvimento *low-code* são um dos maiores segmentos de mercado, com estimativa de que, até 2025, serão responsáveis por mais de 70% do desenvolvimento de aplicativos, conforme Gartner (2021). Essas plataformas são ambientes virtuais que permitem aos usuários a criação de aplicativos para diversos dispositivos sem a necessidade de conhecimentos avançados em linguagens de programação, possibilitando um processo de desenvolvimento intuitivo.

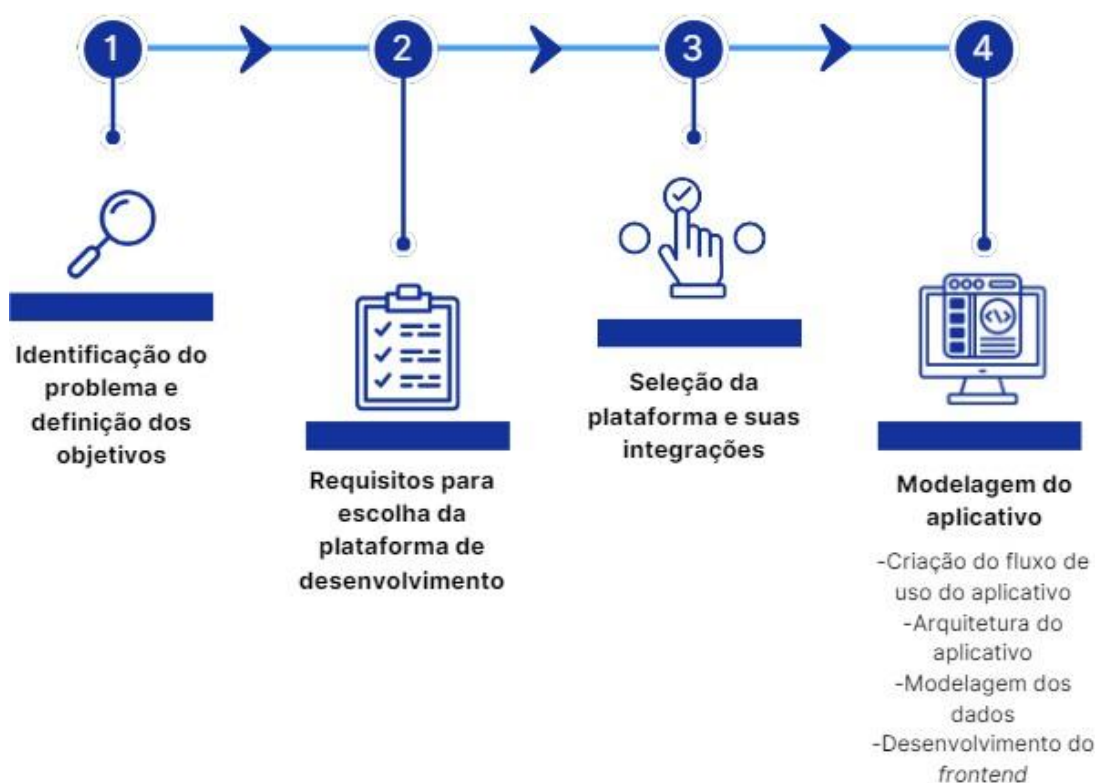
O *low-code* é uma abordagem com programação baseada em modelagem visual utilizando elementos prontos, o que acelera a criação e reduz o tempo e os custos de implementação (Makadia, 2024). Dessa forma, as plataformas *low-code* oferecerem serviços de desenvolvimento de *softwares* com mínimo de definição de código manual, por meio de um ambiente com componentes pré-configurados, o que aumenta o interesse por essas ferramentas virtuais. Além disso, é possível integrá-las a outros serviços, como banco de dados, permitindo a criação de aplicativos desde os mais simples até os mais robustos (Cronapp, 2023).

A utilização de plataformas de desenvolvimento *low-code* são ótimas opções para a criação de versões simples de aplicativos em fase de testes e análise de viabilidades, por oferecer a experiência de um processo mais rápido e focado na validação da ideia. O MVP (*Minimum Viable Product*) é o termo que descreve esse tipo de aplicação, em que se busca identificar uma lacuna em um processo e propor uma solução funcional através de uma construção rápida e ágil do produto.

3 METODOLOGIA DA PESQUISA

Neste capítulo serão abordados os métodos utilizados para o desenvolvimento do aplicativo, apontando as etapas desde a concepção até sua modelagem. As etapas do procedimento metodológico consistem em: Identificação do problema e definição dos objetivos, Requisitos para a escolha da plataforma de desenvolvimento, Seleção da plataforma e suas integrações e Modelagem do aplicativo, conforme ilustrado a seguir.

Figura 6 - Fluxograma metodológico



Fonte: Elaborado pelo autor.

3.1 IDENTIFICAÇÃO DO PROBLEMA E DEFINIÇÃO DOS OBJETIVOS

A utilização da FMEA na atividade de inspeção predial torna o processo lento, principalmente, quando se tem um grande volume de informações. Isso é devido à necessidade de alimentar planilhas e organizar os dados coletados, com base nos

valores do NPR, para a realização das análises. Nesse sentido, sintetizá-la em uma aplicação móvel, facilitaria o processamento das informações coletadas.

O principal objetivo do aplicativo é otimizar a etapa de hierarquização de falhas e organização das informações. Além disso, contribuir para a criação de históricos que auxiliarão na elaboração de planos de manutenção, no monitoramento e na análise em relação à eficácia das ações determinadas. Dessa forma, a ferramenta tem como público-alvo os responsáveis técnicos, engenheiros e arquitetos encarregados pela inspeção predial.

3.2 REQUISITOS PARA A ESCOLHA DA PLATAFORMA DE DESENVOLVIMENTO

Para o desenvolvimento do aplicativo, buscou-se uma plataforma que permitisse sua criação de forma intuitiva e eficaz. Dessa maneira, a escolha da abordagem *low-code* para a criação do aplicativo se fundamentou na necessidade de atender requisitos funcionais e não funcionais, focando na rapidez e facilidade. Os requisitos funcionais são aqueles se referem às funções que a ferramenta deve executar para atender as demandas necessárias. Enquanto, os requisitos não funcionais referem-se ao desempenho, acessibilidade e usabilidade do aplicativo.

Sendo assim, os requisitos funcionais exigidos para o êxito no desenvolvimento da ideia de implementação da metodologia FMEA em um aplicativo são:

- Armazenamento e gerenciamento de dados através da integração com uma plataforma *backend*;
- Manipulação de dados, ou seja, criar, ler, atualizar e deletar (CRUD – *Create, Read, Update, Delete*);
- Cadastro e autenticação do usuário;
- Listagem e exibição de dados;
- Cálculo automático do NPR e organização em ordem decrescente dos dados inseridos;
- Inserir registros fotográficos;
- Verificar o sucesso das ações corretivas propostas;
- Gerar PDF a partir dados inseridos;

Já em relação aos requisitos não funcionais, buscou-se atender aos itens descritos a seguir:

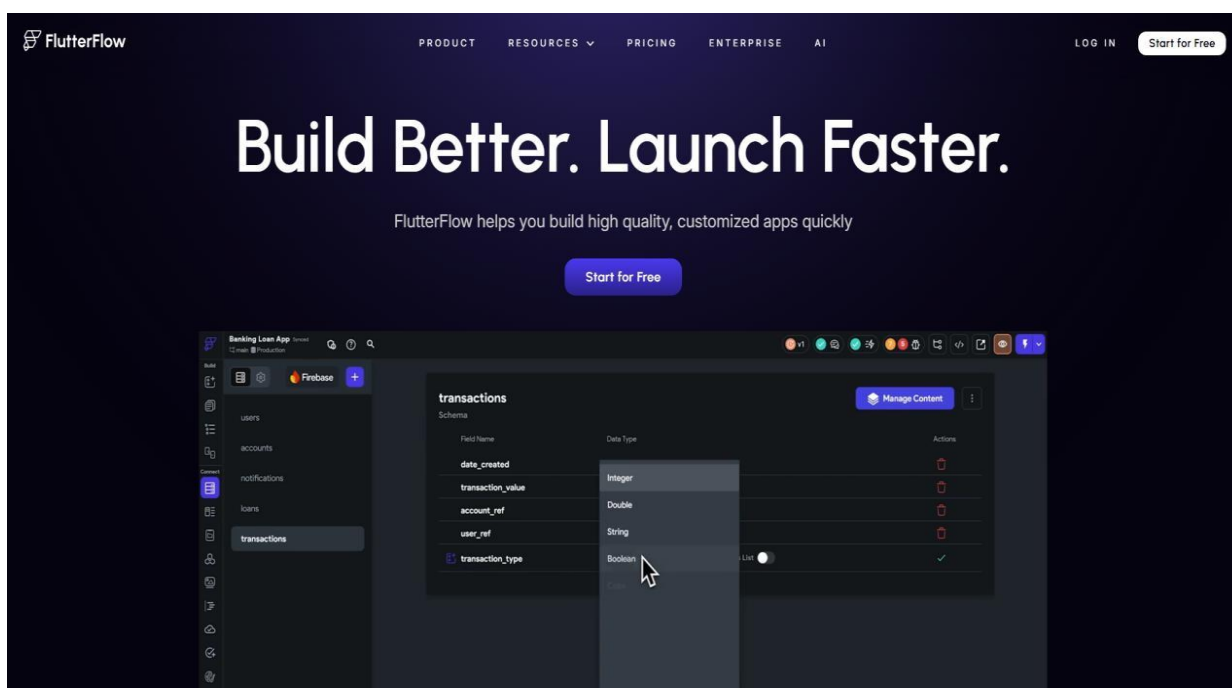
- Compatibilidade com sistemas operacionais como iOS e Android;
- Interface intuitiva e responsiva, ou seja, adaptável aos diferentes tipos de telas;
- Escalabilidade de acordo com aumento de usuários;
- Capacidade de migração de dados;
- Manutenibilidade com a possibilidade de expansão e customização de funções conforme as necessidades.

3.3 SELEÇÃO DA PLATAFORMA E SUAS INTEGRAÇÕES

Com base nas funcionalidades desejadas, a plataforma de desenvolvimento *low-code* selecionada foi o *FlutterFlow* com integração no banco de dados do Supabase para a criação do aplicativo denominado MANiFE, cujo nome é a relação entre a palavra manifestação e FMEA.

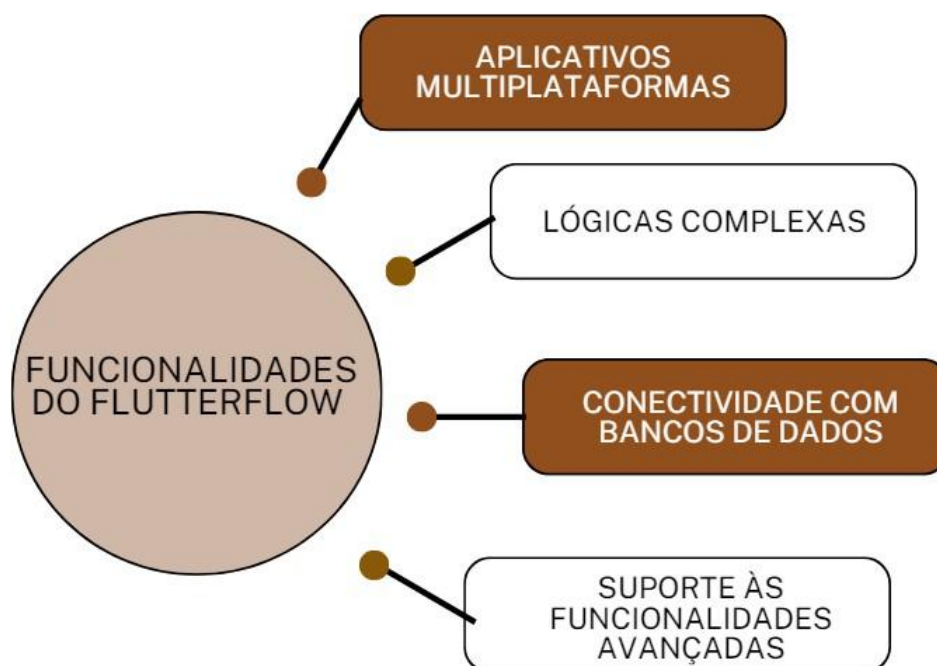
3.3.1 *FlutterFlow*

O *FlutterFlow* é uma ferramenta *low-code* utilizada para o desenvolvimento do *frontend* (parte visual) de aplicativos nativos, isto é, que permitem o acesso a todos os recursos do celular, como por exemplo, câmera, galeria de fotos, entre outros. Essa plataforma proporciona a criação de *apps* com baixo código e de alta qualidade para *web*, iOS e Android, a partir de uma diversidade de componentes prontos em uma interface gráfica de arrastar e soltar (FlutterFlow, 2024). Apesar de oferecer um plano pago, a versão gratuita atende bem como suporte para o desenvolvimento de protótipos para avaliação da viabilidade das aplicações.

Figura 7 - Página inicial do site *Flutterflow*

Fonte: Elaborado pelo autor.

O desenvolvimento dessa plataforma foi baseado no *Flutter*, um *framework* do Google que utiliza a linguagem de programação Dart. Assim, dentro da plataforma, por meio dessa linguagem é possível customizar códigos para criar lógicas e ações de acordo com a complexidade das funcionalidades do aplicativo. O *FlutterFlow* proporciona também o acesso ao código completo do aplicativo e através do plano pago, é possível exportá-lo para editores de código-fonte, como o *Visual Studio Code*. Outra funcionalidade que pode ser executada dentro do *Flutterflow* é a conectividade com bancos de dados nativos, o *Firebase* ou o *Supabase*, visando à manipulação e o armazenamento de informações e funcionalidades mais avançadas, como autenticação de usuários.

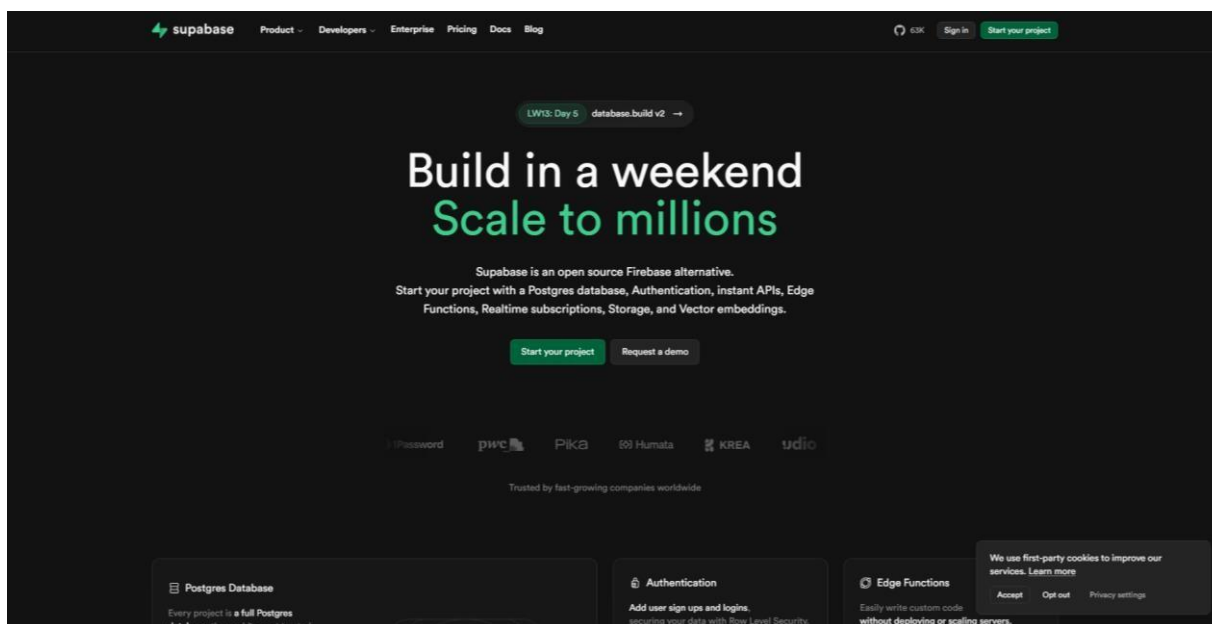
Figura 8 - Funcionalidades do *FlutterFlow*

Fonte: Elaborado pelo autor.

3.3.2 Supabase

O Supabase é uma plataforma *backend* para gestão de banco de dados relacionais com serviços de autenticação, segurança, armazenamento de mídias e funções de API, ou seja, comunicação entre dois sistemas (Supabase, 2024). Essa plataforma utiliza o banco de dados *PostgreSQL*, que oferece escalabilidade ao sistema, assim mesmo com o aumento na quantidade de dados, o seu desempenho é mantido. O Supabase é uma opção bastante atrativa aos desenvolvedores devido às suas regras e boas práticas de proteção e segurança dos dados. Além disso, oferece em seu plano gratuito o armazenamento de 500 MB para o banco de dados e 1GB para arquivos, que é suficiente para projetos em fase inicial.

Figura 9 - Página inicial do site Supabase



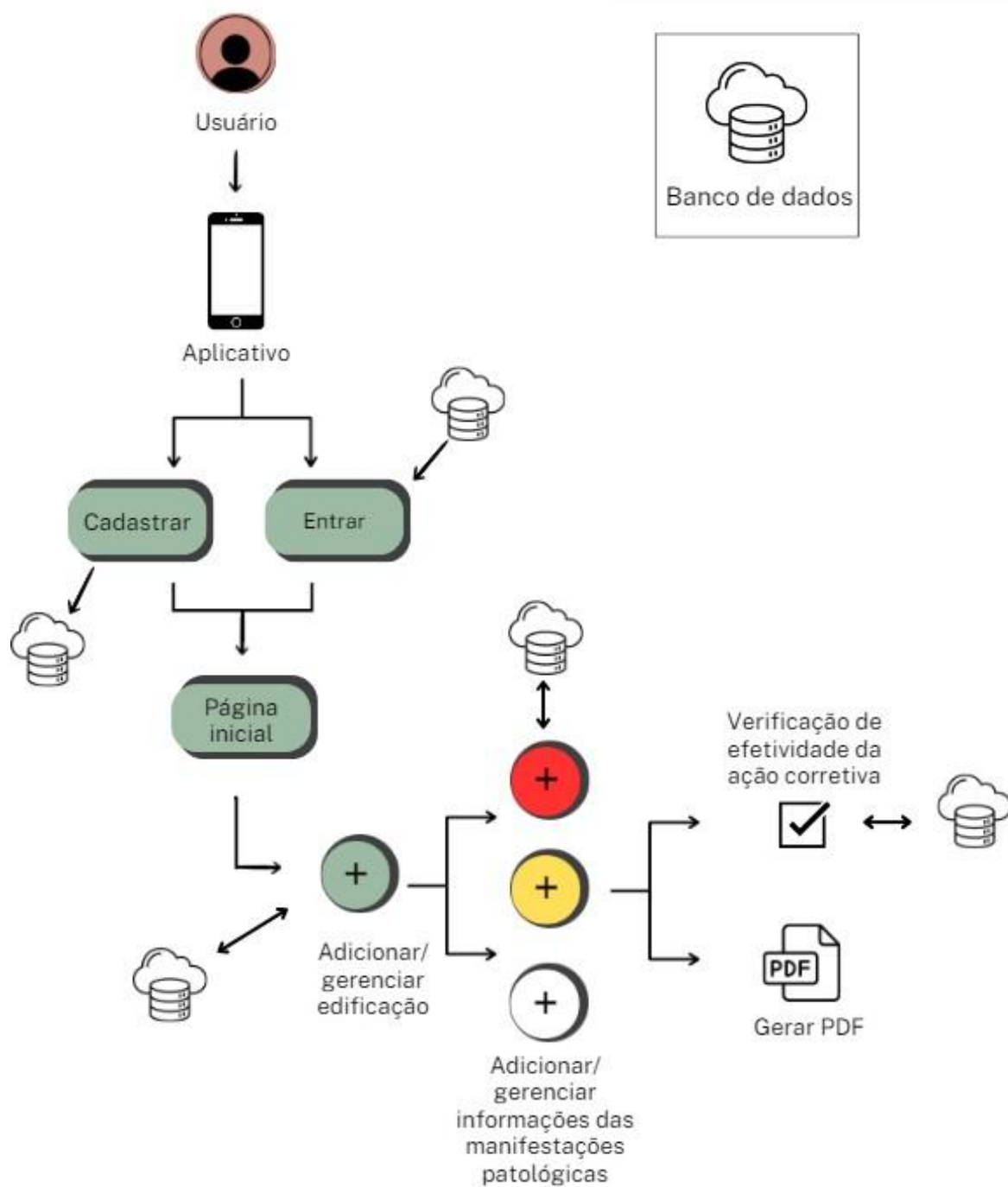
Fonte: Elaborado pelo autor.

3.4 MODELAGEM DO APLICATIVO

3.4.1 Criação do fluxo de uso do aplicativo

Casos de uso ou fluxo de uso trata-se de um diagrama que apresenta as informações relacionadas ao usuário e suas interações com o sistema (Sommerville, 2011). Essa etapa possibilita a descrição e visualização do funcionamento dos aplicativos para auxiliar na estruturação da sequência de telas e como os dados serão tratados. Para o desenvolvimento do MANiFE, a interação entre o usuário e o aplicativo inicia-se no momento da inserção ou cadastro das informações de acesso, direcionando-o para a tela inicial, onde poderá cadastrar os dados do edifício. Em cada edifício cadastrado, será possível inserir as manifestações patológicas observadas e as ações corretivas propostas pelo responsável da inspeção. Essas informações serão exibidas de forma hierarquizadas em ordem decrescente, conforme o valor do NPR calculado. Por fim, o usuário poderá ainda gerar o documento PDF. Todas as informações inseridas pelo usuário serão armazenadas em um banco de dados, facilitando o gerenciamento das informações dentro do aplicativo, como editar ou excluir. A Figura 10 mostra todo o fluxo de uso do aplicativo.

Figura 10 - Fluxo de uso do aplicativo

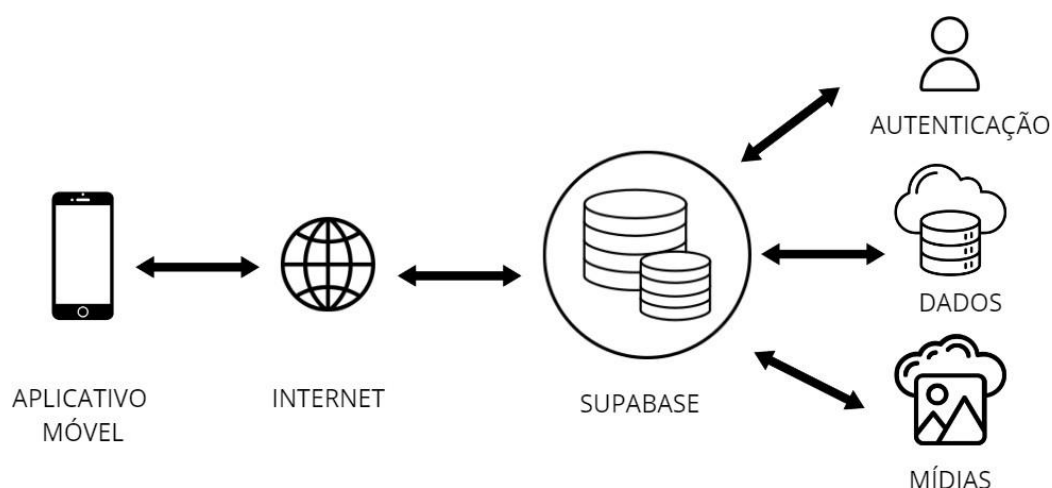


Fonte: Elaborado pelo autor.

3.4.2 Arquitetura do aplicativo

A arquitetura de um projeto de *software* é uma forma criativa e organizada de representar através de diagramas de blocos simples, um projeto que busca atender aos requisitos funcionais e não funcionais de um sistema (Sommerville, 2011). O objetivo é melhorar o entendimento da interação entre os componentes de um sistema e guiar seu processo criativo. No Supabase, os principais serviços utilizados para atender os requisitos do aplicativo, com base no fluxo de uso estabelecido, são: *PostgreSQL database* para o gerenciamento e relacionamento dos dados, o *Storage* para o armazenamento de fotos e *Authentication* para controle de acessos.

Figura 11 - Arquitetura de aplicação



Fonte: Elaborado pelo autor.

3.4.3 Modelagem dos dados

O Supabase possibilita a modelagem de dados por meio do relacionamento entre tabelas. Dessa forma, inicialmente, estabeleceu-se quais dados seriam armazenados e a lógica de fluxo das informações de acordo com a previsão do modo de funcionamento do aplicativo. As possibilidades de relacionamento entre tabelas dentro do Supabase são: um para um (1:1), um para muitos (1:n) e muitos para muitos (n:n).

Essa plataforma já oferece uma tabela padrão para a autenticação do usuário, onde armazena senhas e e-mails. No entanto, é necessário criar a tabela de usuário

do aplicativo e estabelecer a conexão entre elas, por meio de uma relação um para um (1:1). No aplicativo, também foi utilizado a relação de um para muitos para o tratamento dos dados entre o usuário, a edificação e as manifestações patológicas. Isso devido ao fato que um usuário pode registrar várias edificações, assim como em uma edificação podem ser adicionadas várias manifestações patológicas.

Conforme as necessidades do aplicativo, foram criadas três tabelas, apresentadas na Tabela 8.

Tabela 8 - Atributos para as tabelas criadas no Supabase

Nome da tabela	Atributos
Usuario	Possui as informações relacionadas ao usuário como Nome, E-mail e o identificador (chave-primária).
Dadosdaedificacao	Possui as informações relacionadas à edificação como identificador (chave primária), Nome da edificação, Nome do responsável técnico, Local da edificação, Data da vistoria e o identificador do usuário que se relaciona com a chave primária da tabela “Usuario”.
Dadosmanifestacaopatologica	Possui as informações relacionadas às manifestações patológicas como a chave primária, Nome da patologia, Local, Foto, Efeito, Causa, Severidade, Ocorrência, Detecção, Ação corretiva e o identificador da edificação que se relaciona com a chave primária da tabela “Dadosdaedificacao”. Além disso, adicionou a coluna referente ao NPR, no qual será calculado conforme os valores de severidade, ocorrência e detecção inseridos pelo usuário. Também foram adicionadas as colunas nomeadas como <i>Check</i> e <i>Status</i> , para a checagem da efetividade da ação corretiva definida.

Fonte: Elaborado pelo autor.

Para o cálculo do NPR, foi inserido a seguinte fórmula no campo *Default value*, demonstrada na Figura 12, para atribuir automaticamente o resultado de acordo com os valores inseridos.

Figura 12 - Preenchimento do campo com a fórmula do NPR

The screenshot shows the Supabase SQL Editor interface for updating a column. The title bar reads "Update column npr from Dadosmanifestacaopatologica". The interface is divided into two main sections: "General" and "Data Type".

General Section:

- Name:** A text input field containing "npr". Below it, a note says: "Recommended to use lowercase and use an underscore to separate words e.g. column_name".
- Description:** An empty text input field. To its right, the word "Optional" is displayed.

Data Type Section:

- Data Type:** On the left, there are two buttons: "+ Create enum types" and "About data types".
- Type:** A dropdown menu showing "# int4".
- Options:**
 - ☐ **Is Identity**: Automatically assign a sequential unique number to the column.
 - ☐ **Define as Array**: Allow column to be defined as variable-length multidimensional arrays.
- Default Value:** A text input field containing "s * o* d". Below it, a note says: "Can either be a literal or an expression. When using an expression wrap your expression in brackets, e.g. (gen_random_uuid())".

Fonte: Elaborado pelo autor.

As colunas referentes à checagem das ações corretivas definidas para o tratamento da manifestação patológica foram criadas mantendo uma relação. Assim, quando no aplicativo for definido o valor *Boolean True* para a coluna "Check", na coluna "Status" será exibido o texto "Resolvido", caso for *False* ou um valor nulo, exibirá "Em andamento". Para isso, através do *SQL Editor* no Supabase, foi inserido o código com a lógica condicional, demonstrada na Figura 13. Esse código utiliza a linguagem declarativa SQL, que permite as operações como a atualização dados (*Update* e *Set*) e aplicações de condições (*Case*), abordadas para esse contexto.

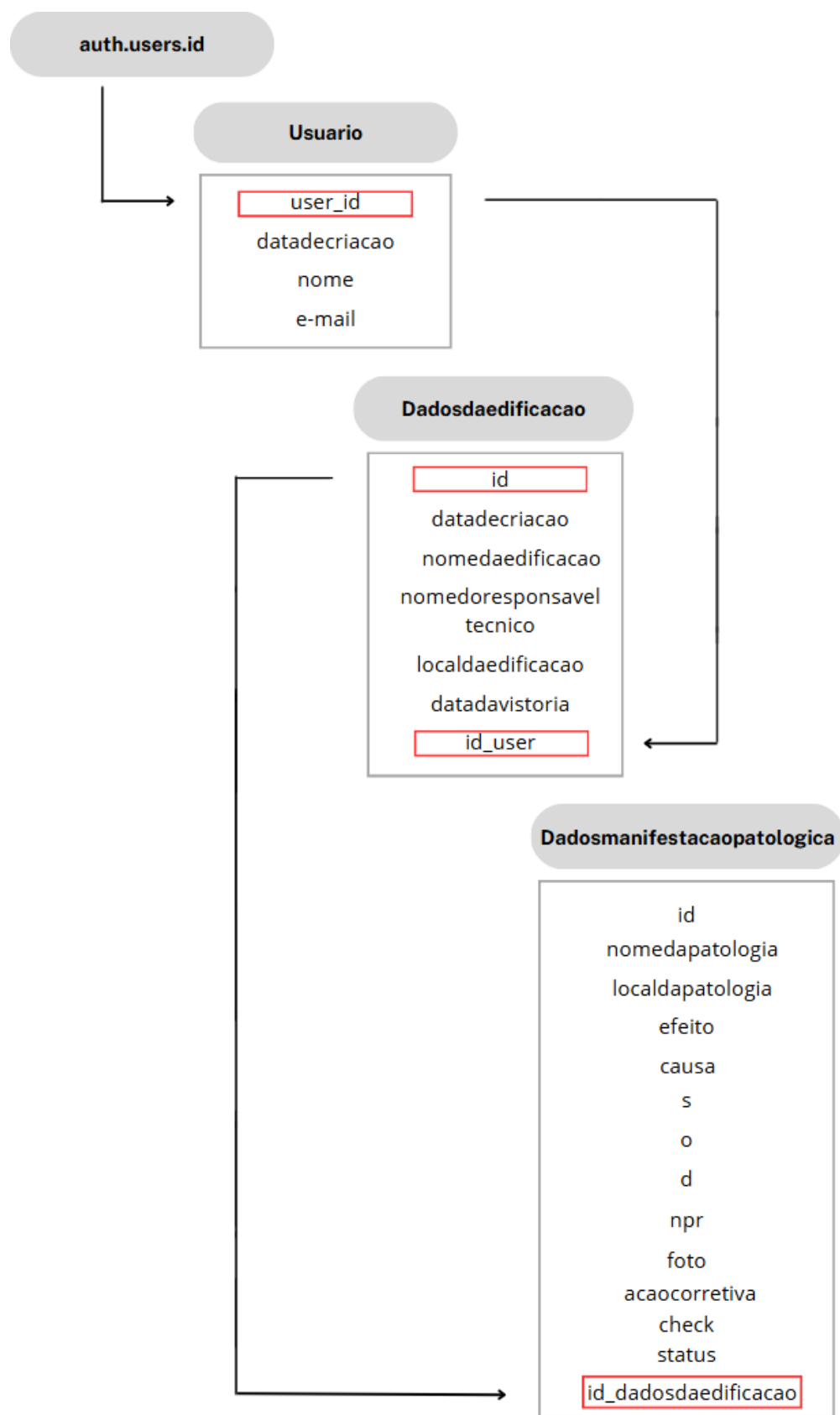
Figura 13 - Código para a condição da coluna "Status" no banco de dados

```
1 UPDATE "Dadosmanifestacaopatologica"
2 SET "status" = CASE
3     WHEN "check" = 'true' THEN 'Resolvido'
4     WHEN "check" = 'false' THEN 'Em andamento'
5     WHEN "check" IS NULL THEN 'Em andamento'
6 END;
7
```

Fonte: Elaborado pelo autor.

As colunas de identificadores primários são automaticamente criadas pelo Supabase, assim como a coluna que se refere à data de criação dos dados, necessária apenas para as tabelas "Usuario" e "Dadosdaedificacao", para facilitar posteriores consultas. Para que as tabelas se relacionem entre si, é preciso criar chaves com ID referentes às outras tabelas, como realizado nas tabelas "Dadosdaedificacao" e "Manifestacaopatologica", conforme a Figura 14.

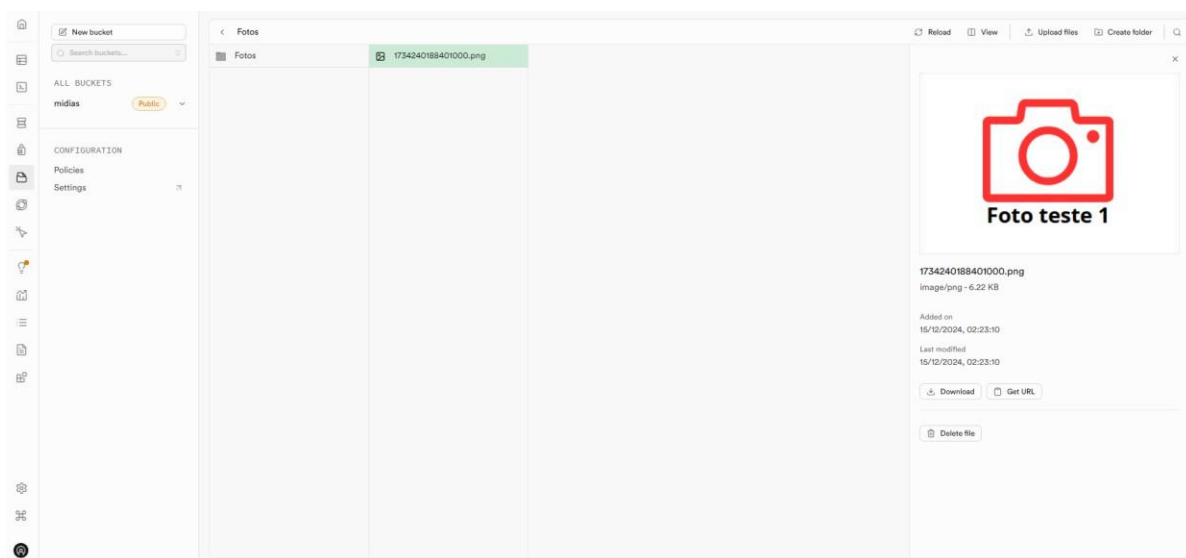
Figura 14 - Esquema da lógica de relacionamento entre as tabelas no Supabase



Fonte: Elaborado pelo autor.

Além disso, utilizando o recurso *Storage*, criou-se um *bucket*, isto é, uma pasta que foi denominada de “mídias” para salvar e gerenciar fotos, como ilustrado na figura abaixo:

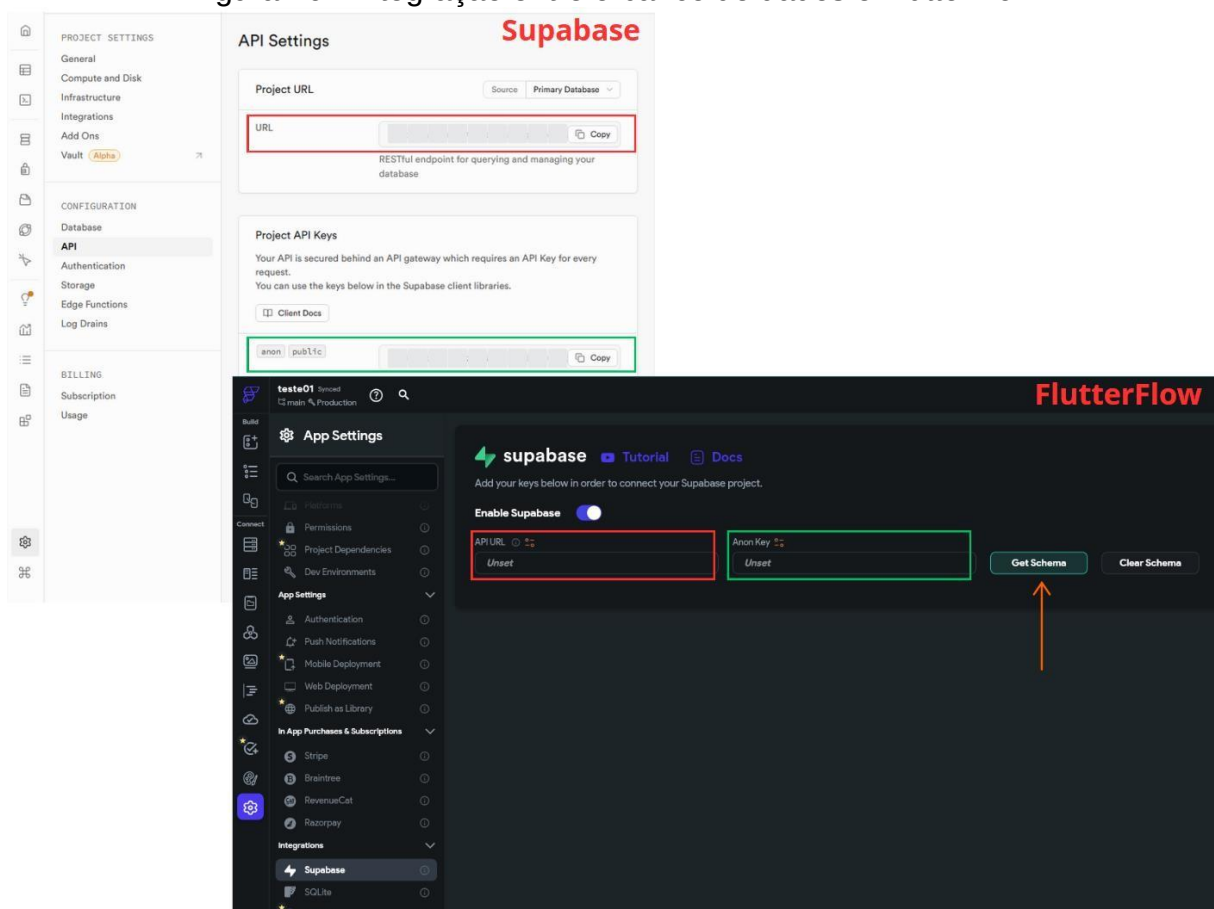
Figura 15 - Pasta de mídias no banco de dados



Fonte: Elaborado pelo autor.

Posteriormente, nas configurações da plataforma de desenvolvimento *FlutterFlow*, integrou-se o banco de dados através da URL da API e a chave *Anon* do projeto criado no Supabase. Para a integração entre as tabelas e o aplicativo, clicou-se no botão *Get Schema*, como está ilustrado na figura a seguir.

Figura 16 - Integração entre o banco de dados e *FlutterFlow*

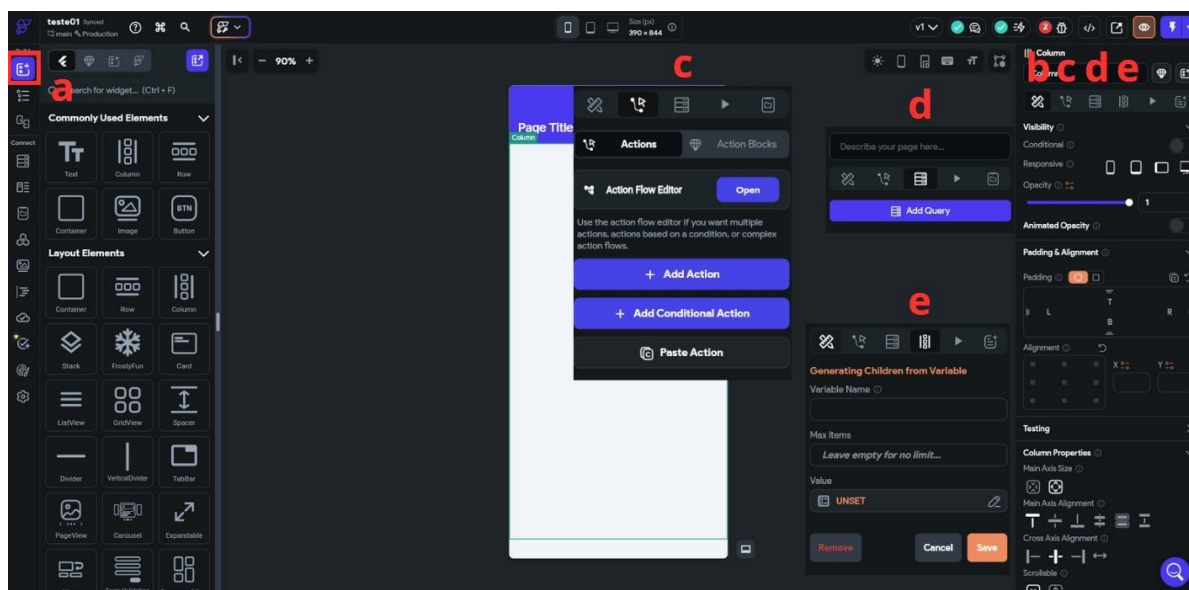


Fonte: Elaborado pelo autor.

3.4.4 Desenvolvimento do *frontend*

As principais funcionalidades da plataforma do *FlutterFlow* utilizadas para o desenvolvimento da interface gráfica e manipulação de dados dentro do aplicativo, foram o *Menu Proprieties*, *Action*, *Backend Query* e *Generate Dynamic Children*. O *Menu Proprieties* (Figura 17, b), são configurações responsáveis pela personalização gráfica dos recursos prontos disponíveis no *Widgets Palltete* (Figura 17, a). Para melhor compreensão, na Tabela 9 estão descritos todos os recursos que foram necessários para a criação das telas do aplicativo.

A funcionalidade *Action* (Figura 17, c) possibilita atribuir ações aos componentes da tela. O *Backend Query* (Figura 17, d) permite a manipulação dos dados ao conectar o elemento da interface a um banco de dados ou API's externa, enquanto o *Generate Dynamic Children* (Figura 17, e) recupera as informações de um banco de dados, gerando listas dinâmicas.

Figura 17 - Área de desenvolvimento da plataforma *FlutterFlow*

Fonte: Elaborado pelo autor.

Tabela 9 - Tabela de descrição dos recursos utilizados para a criação das telas

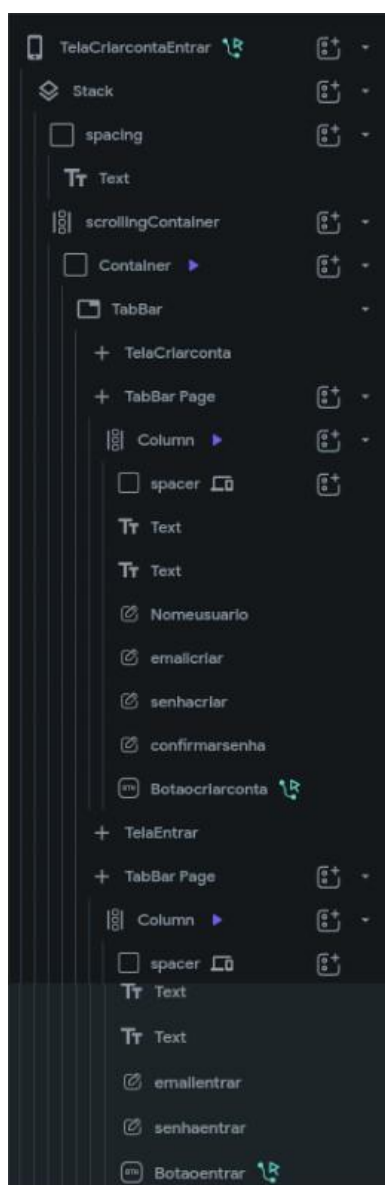
Widget	Descrição
<i>Container</i>	Organiza componentes menores e estiliza a tela
<i>Column</i>	Organiza em sequência vertical os componentes
<i>Rows</i>	Organiza os componentes lado a lado
<i>Listview</i>	Lista de navegação vertical ou horizontal de componentes
<i>Form validation</i>	Adiciona validação aos campos de entrada de dados
<i>Datatable</i>	Usado para exibir dados no formato de tabelas
<i>TabBar</i>	Exibe uma linha horizontal de abas
<i>AppBar</i>	Exibe uma barra no topo da tela para navegação
<i>Drawer</i>	Usado para exibir um menu de navegação deslizante
<i>Image</i>	Usado para exibir imagens
<i>Icon</i>	Usado para exibir símbolos de ícones
<i>Button</i>	Usado como meio para executar ações
<i>Checkbox</i>	Caixa de seleção usado para capturar escolhas de usuários, como verdadeiro/falso ou sim/não

Fonte: Elaborado pelo autor.

3.4.4.1 Tela Criar conta/Entrar

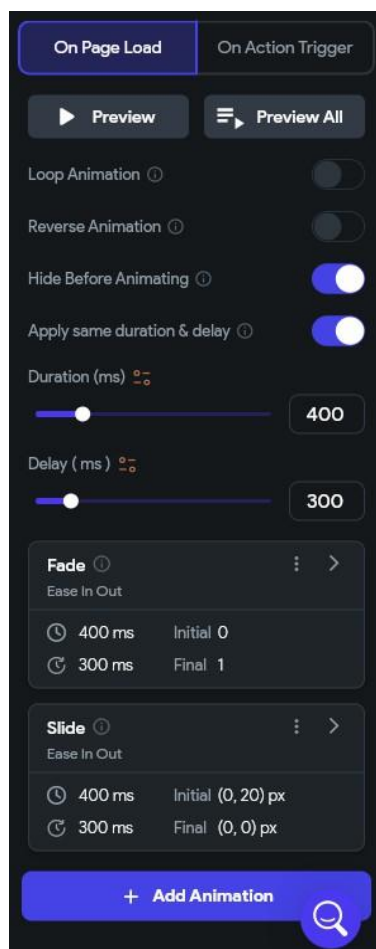
O desenvolvimento da interface do aplicativo iniciou-se pela tela Criar conta/Entrar, cujo o objetivo é autenticar o acesso dos usuários. Nessa tela, para o cadastro do usuário serão apresentados os campos Nome, E-mail e Criar senha, enquanto para entrar, apenas os campos E-mail e Senha (Figura 18). As setas indicam ações e animações que foram atribuídas aos componentes. As animações foram adicionadas para permitir que o usuário navegue, dentro da mesma tela, entre os formulários Criar conta e Entrar (Figura 19).

Figura 18 - Estrutura da tela Criar conta/ Entrar



Fonte: Elaborado pelo autor.

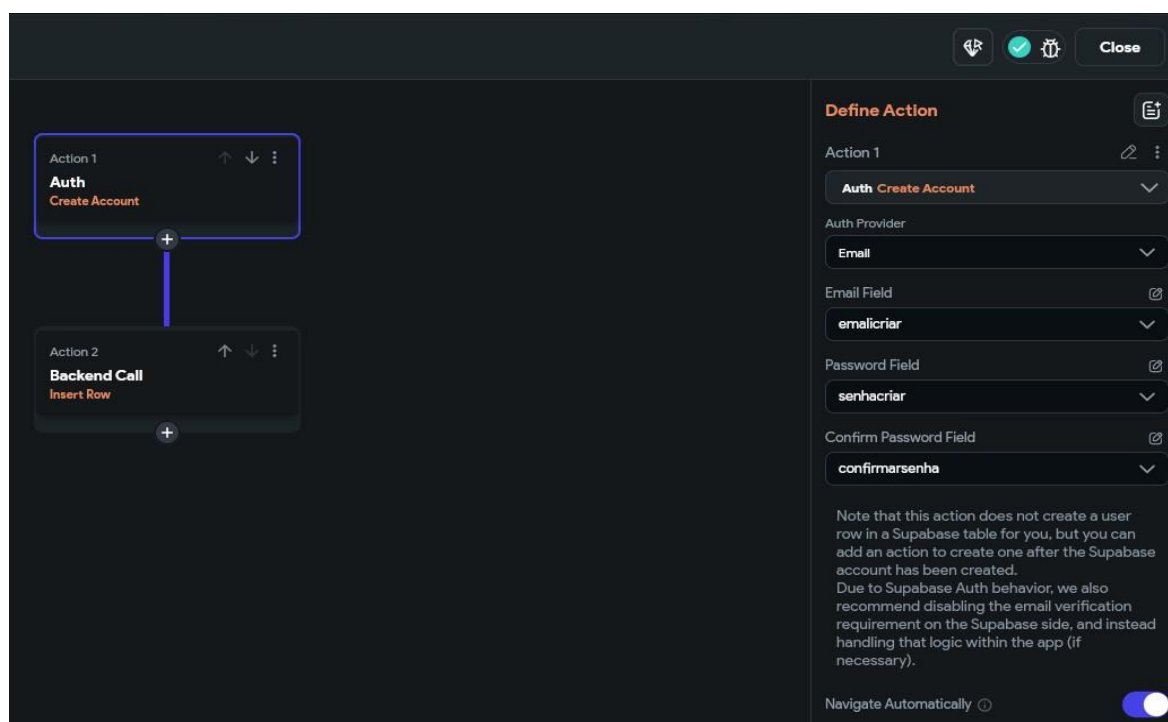
Figura 19 - Animação atribuída à tela



Fonte: Elaborado pelo autor.

As ações atribuídas ao botão Criar conta estão descritas na Figura 20. A primeira ação consiste na autenticação das informações requeridas para a criação da conta. A segunda ação encaminha os dados ao Supabase, por meio da funcionalidade *Backend Call: Insert Rows*, que insere linhas na tabela “Usuario” no banco de dados com as informações preenchidas nos campos.

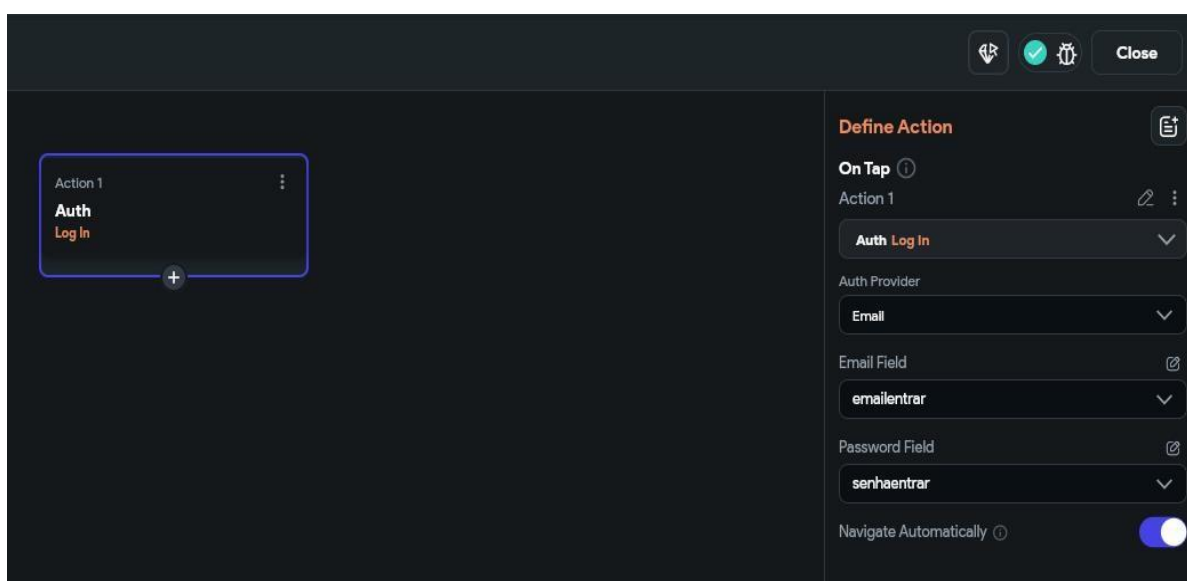
Figura 20 - Sequência de ações para o botão Criar conta



Fonte: Elaborado pelo autor.

Para que o usuário acesse uma conta já criada no aplicativo, é necessário apenas ação de autenticação, realizada por meio do botão Entrar, após o preenchimento das informações requeridas para o acesso às funções do aplicativo.

Figura 21 - Sequência de ações para o botão Entrar

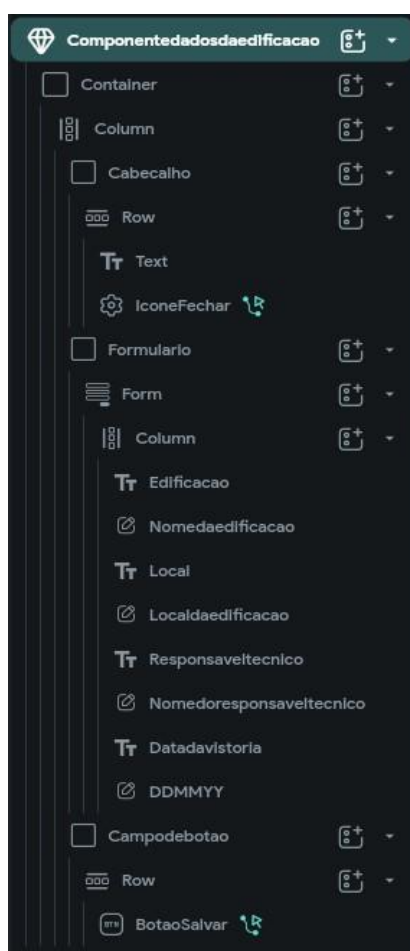


Fonte: Elaborado pelo autor.

3.4.4.2 Formulários

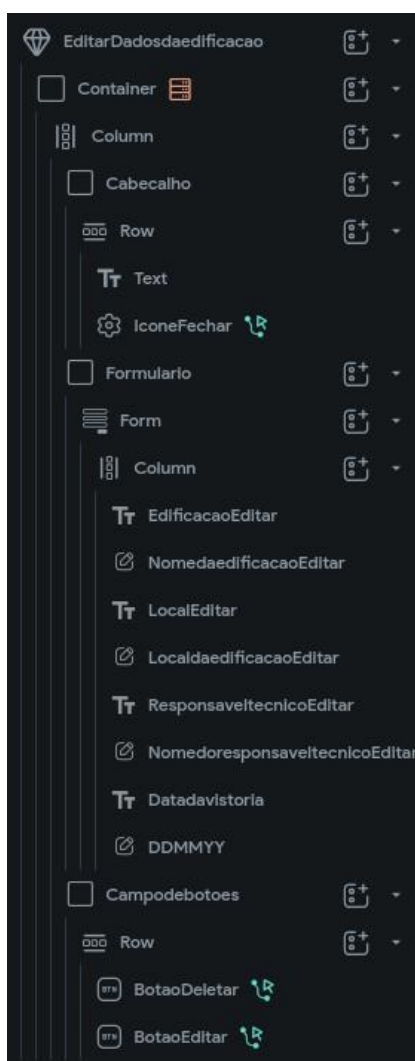
Para a construção dos formulários, foram utilizados os componentes que consistem em *widgets* reutilizáveis, facilitando a padronização e a manutenção (Flutterflow, 2024). Foram criados quatro componentes com os campos necessários para o tratamento das informações no aplicativo. Primeiramente, desenvolveu-se o formulário relacionado aos dados da edificação inspecionada, contendo os campos Edificação, Local, Responsável técnico e a Data da vistoria. Ao salvar, essas informações são armazenadas no banco de dados e exibidas na tela. Esse componente foi reutilizado para permitir ao usuário editar ou excluir registros. No entanto, nesse caso, foi necessário conectar os campos à tabela “Dadosdaedificacao” do banco de dados para possibilitar a visualização das informações já preenchidas. As estruturas dos componentes são apresentadas a seguir.

Figura 22 - Estrutura do componente de entrada dos dados da edificação



Fonte: Elaborado pelo autor.

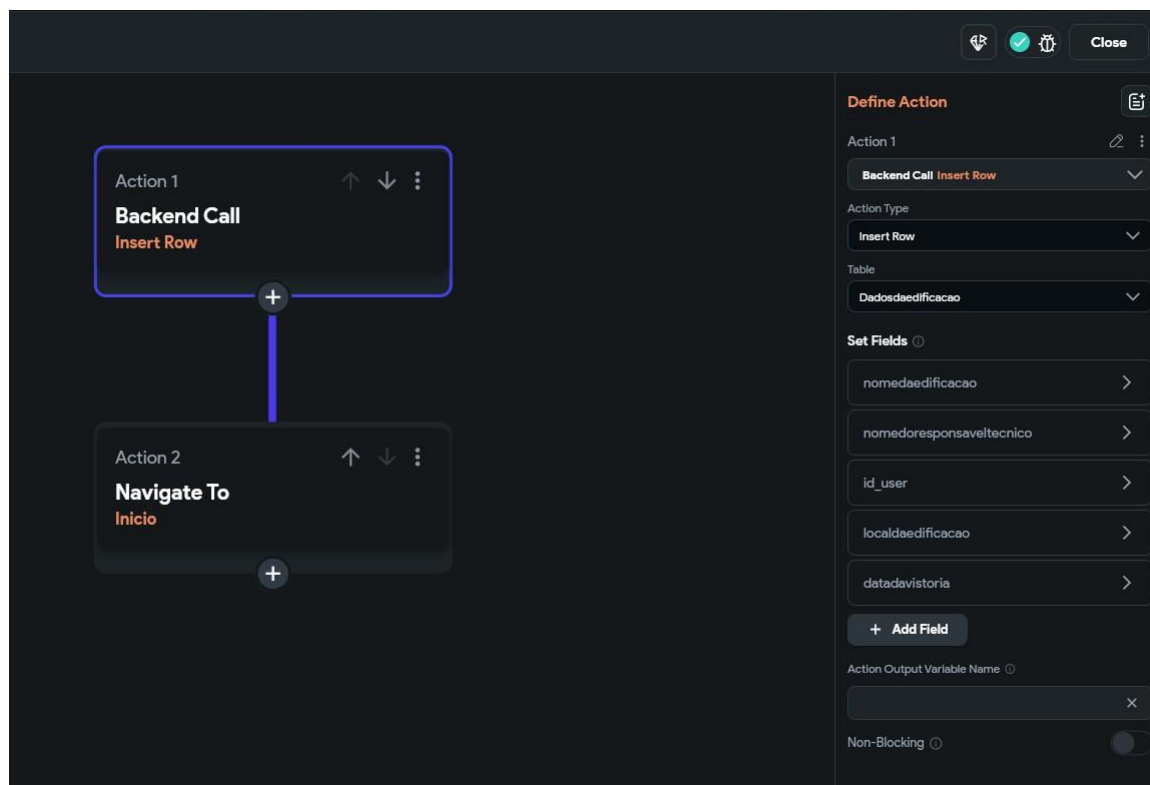
Figura 23 - Estrutura da componente de editar ou deletar os dados da edificação



Fonte: Elaborado pelo autor.

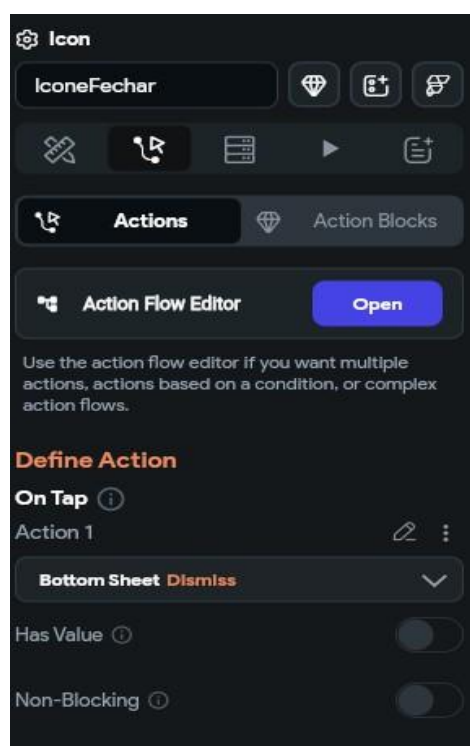
Nas Figura 24 e 25 estão representadas as ações que foram definidas para o botão de salvar as informações e fechar o componente, respectivamente. Para o botão Salvar, o processo é o mesmo descrito na seção 3.4.4.1, em que através da ação *Backend Call: Insert Rows*, as informações preenchidas são salvas na tabela do banco de dados, que nesse caso, é a “Dadosdaedificacao”. Para o botão fechar, foi definida a função *Dismiss*.

Figura 24 - Sequência de ações para o botão Salvar do componente dos dados da edificação



Fonte: Elaborado pelo autor.

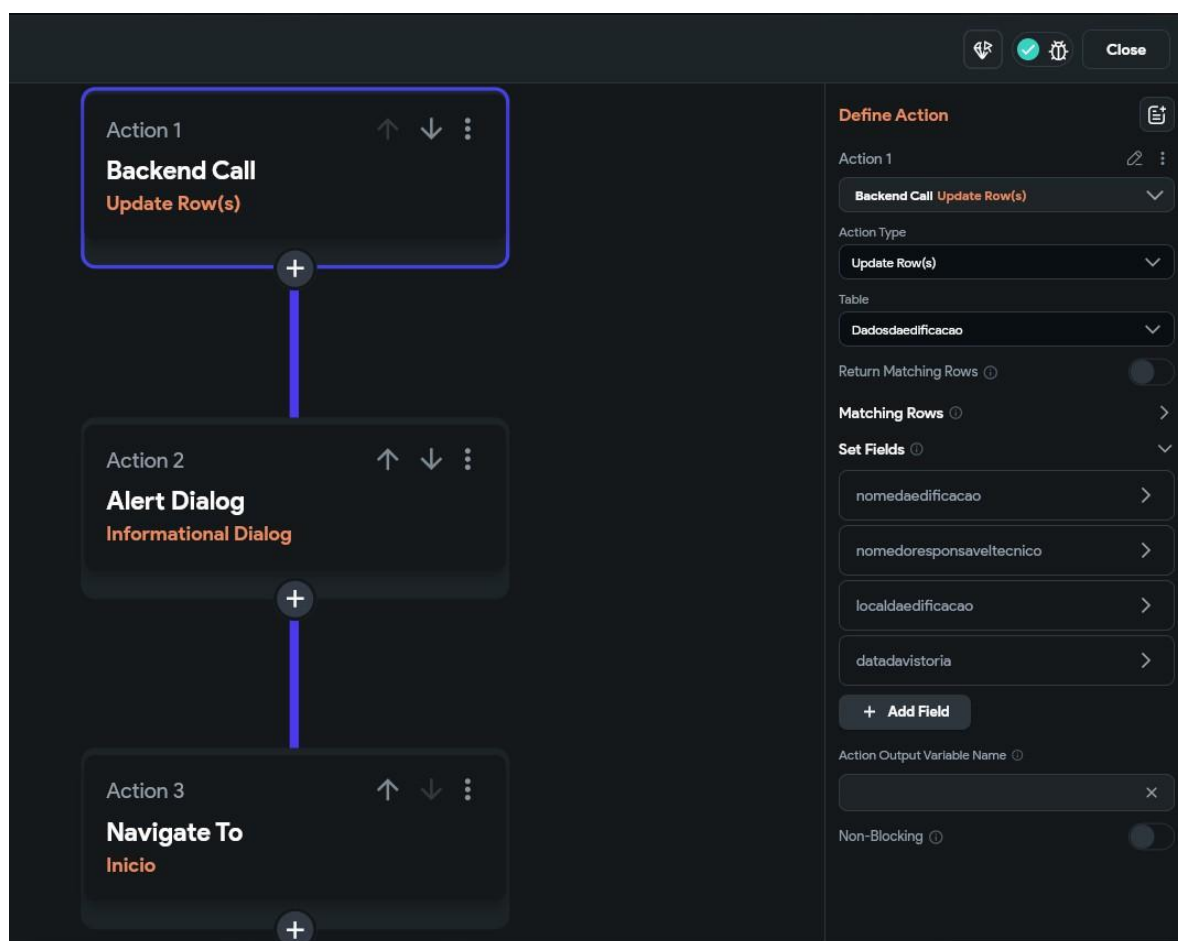
Figura 25 - Ação para o botão Fechar do componente dos dados da edificação



Fonte: Elaborado pelo autor.

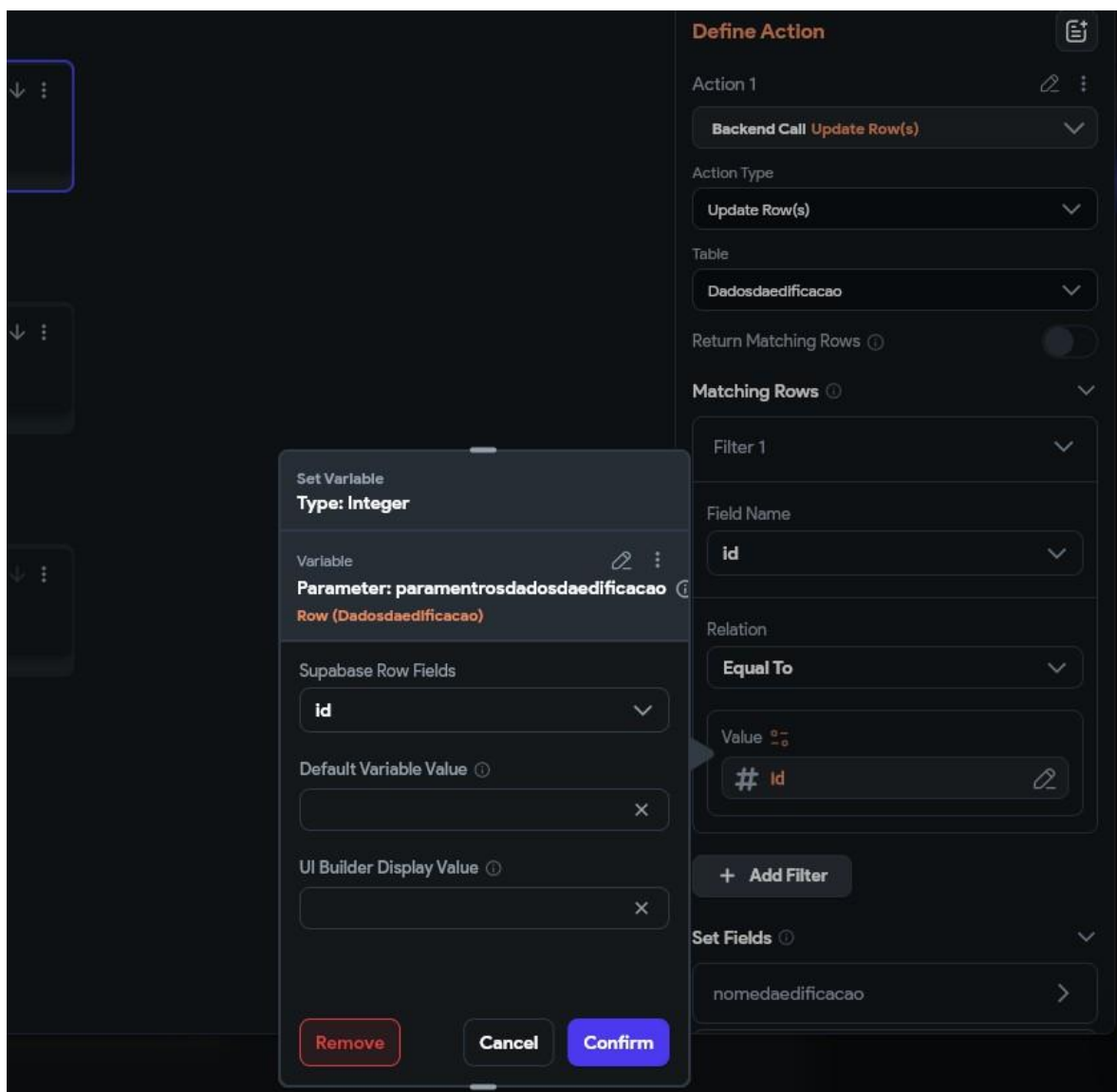
Já para o botão de editar (Figura 23), utilizou-se a ação *Backend Call: Update Rows*, na qual, ao invés de inserir, os dados são atualizados no Supabase (Figura 26). Também foi necessário a funcionalidade *Matching Rows*, com a relação de igualdade entre o ID da tabela “Dadosdaedificacao” e o ID do parâmetro de tela criado para que as informações sejam filtradas e exibidas corretamente para cada usuário logado, como mostrado na Figura 27. Em seguida, foi adicionada a exibição de uma mensagem indicando o sucesso da operação, fechando o componente de edição.

Figura 26 - Sequência de ações para o botão Editar do componente dos dados da edificação



Fonte: Elaborado pelo autor.

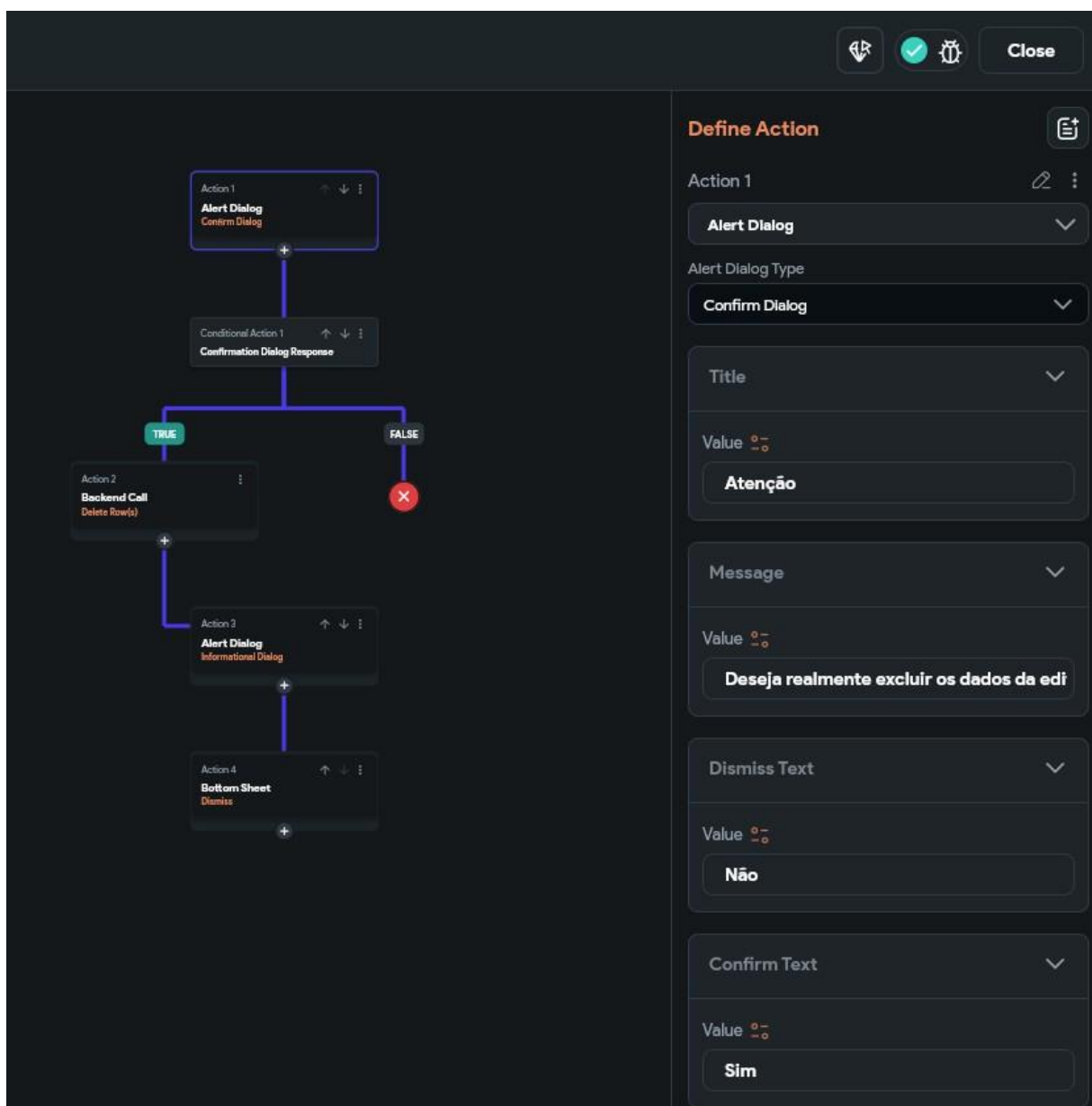
Figura 27 - Filtro para a busca e exibição dos dados



Fonte: Elaborado pelo autor.

Já para o botão Deletar (Figura 23), foi configurada uma caixa de confirmação para a ação. Assim, na situação em que o usuário negue a ação de excluir, o componente se fechará. O contrário, as informações serão excluídas utilizando a ação *Backend Call: Delete Rows*, que deletará os dados do Supabase e da tela do aplicativo. Dentro dessa ação também foi definido o mesmo filtro aplicado anteriormente na funcionalidade *Matching Rows*. Após a execução do comando de exclusão, o componente será fechado, exibindo uma mensagem de sucesso na operação realizada. A sequência lógica das ações está detalhada na Figura 28.

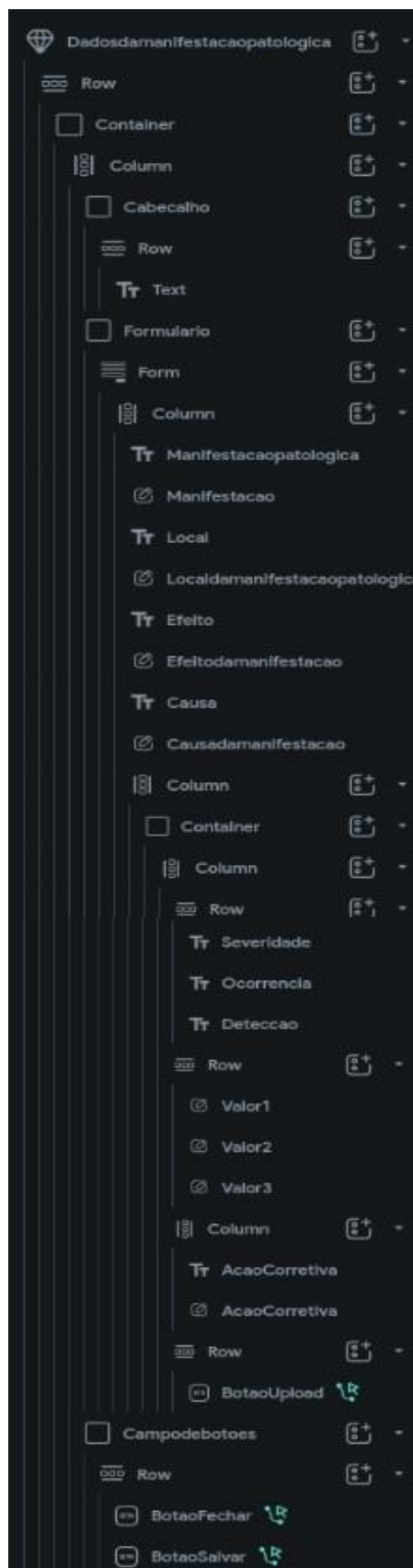
Figura 28 - Sequência de ações para o botão Deletar do componente dados da edificação



Fonte: Elaborado pelo autor.

Após a elaboração dos componentes relacionados aos dados da edificação, criou-se o componente do formulário para os dados das manifestações patológicas. Esse formulário contém os seguintes campos: Manifestação patológica, Local, Efeito, Causa, Severidade, Ocorrência, Detecção, Ação corretiva, além de um botão para *upload* de fotos. A estrutura desse componente é apresentada na Figura 29.

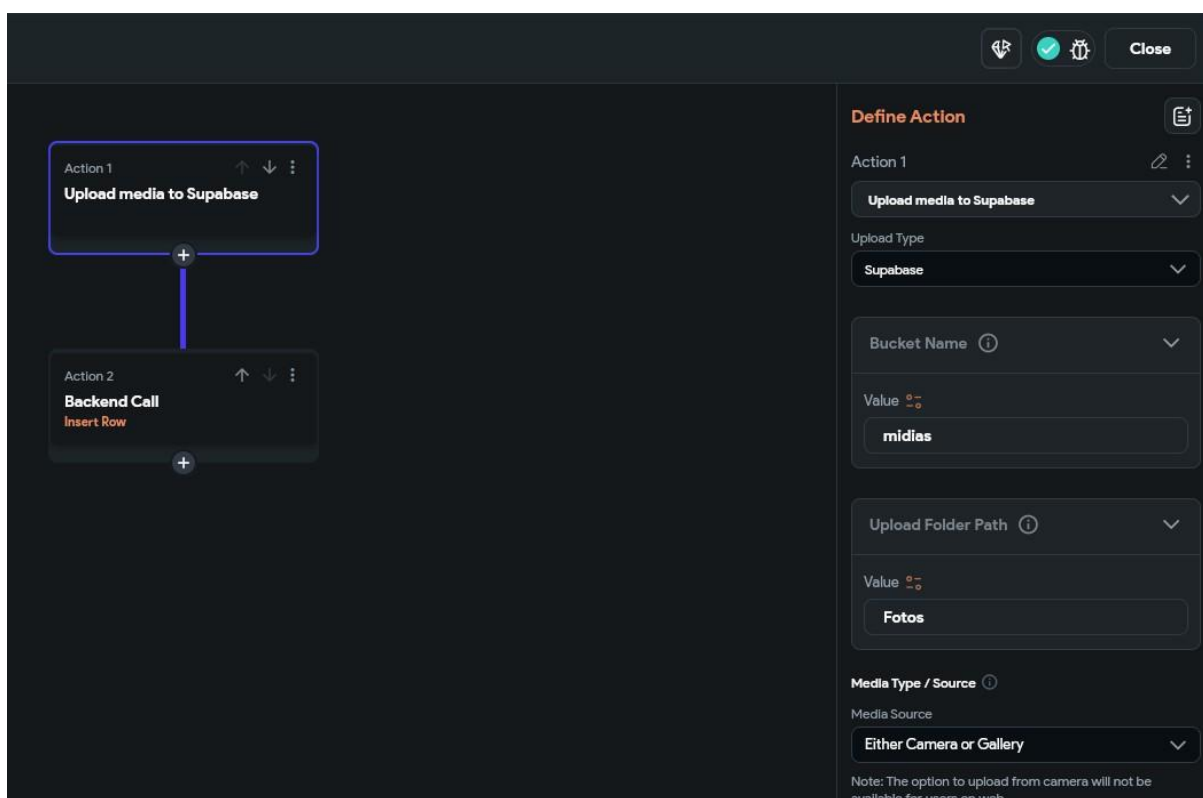
Figura 29 - Estrutura do componente de entrada de dados da manifestação patológica



Fonte: Elaborado pelo autor.

Da mesma forma que no formulário dos dados da edificação, conectou-se os campos de preenchimento com tabela referente às manifestações patológicas e através da ação *BackendCall: Insert Rows*, atribuída ao botão Salvar, as informações são inseridas no banco de dados. Para o botão Fechar também foi atribuído a ação *Dismiss*. Já para realizar o *upload* da mídia do celular do usuário, foi necessário definir a ação de *Upload media to Supabase*, para que a foto da câmera ou da galeria do celular seja carregada para a pasta de mídias no banco de dados. Essa informação é também inserida na tabela e exibida na tela do aplicativo. A sequência de ações está ilustrada abaixo:

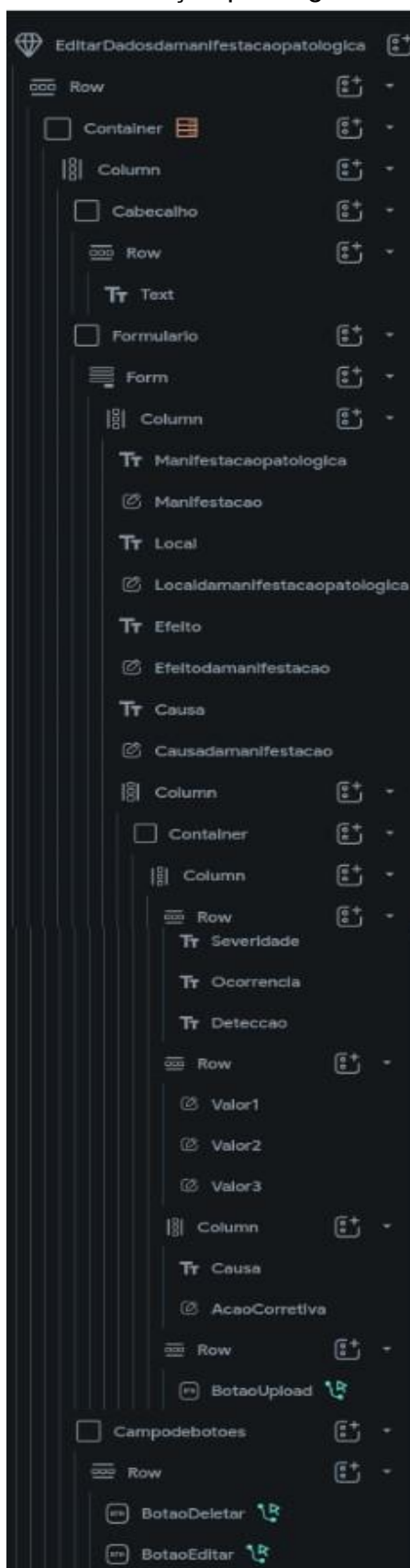
Figura 30 - Sequência de ações para o *upload* da foto



Fonte: Elaborado pelo autor.

As funções de editar e deletar também utilizaram a mesma estrutura do componente de entrada de dados das manifestações patológicas, com mudanças apenas nos botões e suas funções. A atribuição de comandos para os botões Editar e Deletar procedeu-se igual à descrita para componente de edição dos dados da edificação, com a diferença apenas na tabela à qual está vinculada no banco de dados, que trata-se das manifestações patológicas.

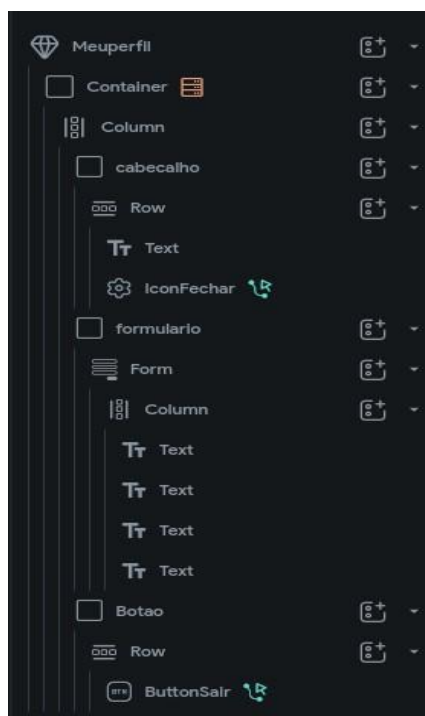
Figura 31 - Estrutura do componente de editar ou deletar os dados da manifestação patológica



Fonte: Elaborado pelo autor.

Outro componente foi criado para a exibição dos dados do usuário logado. Na Figura 32 é apresentado a estrutura do componente Meu perfil.

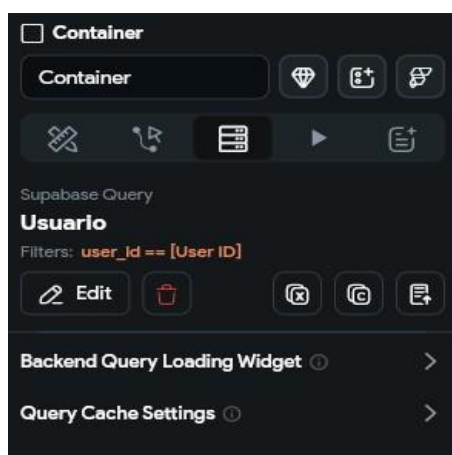
Figura 32 - Estrutura do componente Meu perfil



Fonte: Elaborado pelo autor.

Ao *container* desse componente, conforme a Figura 33, foi adicionada a tabela “Usuario” por meio do *Supabase Query*, que é filtrada pelo “User ID” da tabela criada automaticamente no banco de dados.

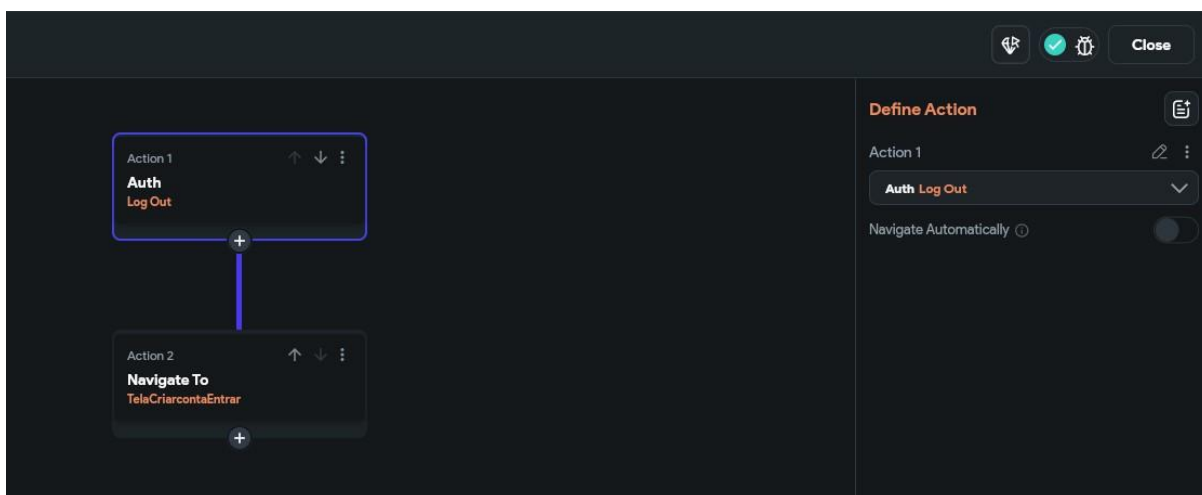
Figura 33 - Atribuição da tabela "Usuario" ao *container* do componente Meu perfil



Fonte: Elaborado pelo autor.

Nesse componente, o usuário tem acesso ao botão Sair. Para a realização desse comando foi atribuída a ação de autenticação *Log Out*, para que o usuário saia da sua conta no aplicativo. Em seguida, o aplicativo retorna à tela inicial de Criar conta ou Entrar.

Figura 34 - Sequência de ações atribuídas ao botão Sair do componente Meu perfil

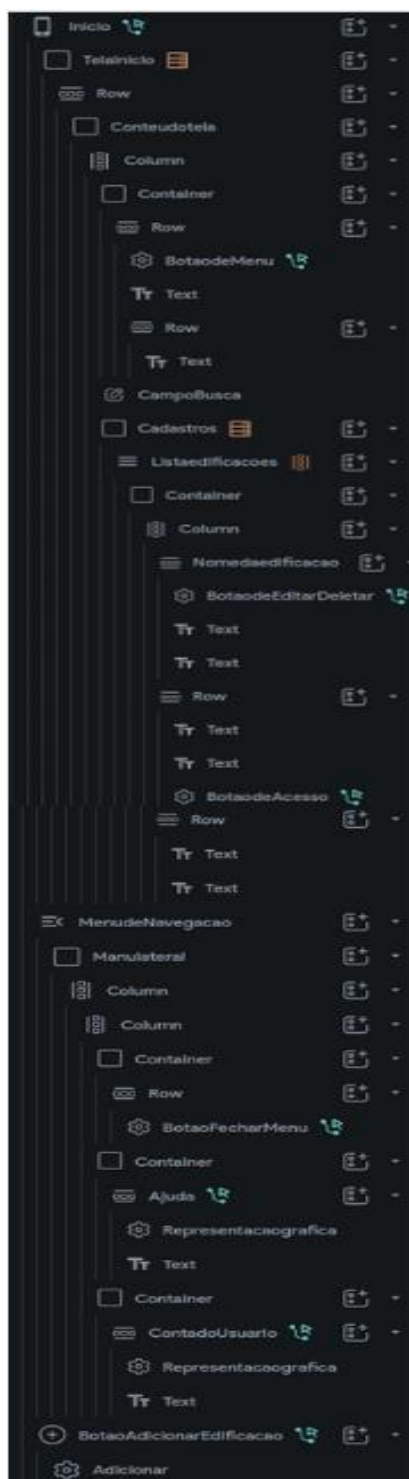


Fonte: Elaborado pelo autor.

3.4.4.3 Tela Início

Na tela Início será possível cadastrar edificações, editar e visualizar as informações cadastradas. A estruturação da interface está ilustrada na Figura 35.

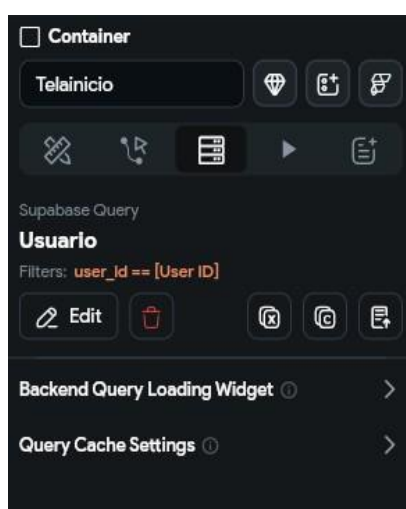
Figura 35 - Estrutura da tela Início



Fonte: Elaborado pelo autor.

No *container* “Telainicio”, foi adicionado filtro para a manipulação dos dados entre as tabelas do banco de dados. Como um usuário pode cadastrar várias edificações, é necessário que o aplicativo entenda e exiba as informações corretamente. Assim, por meio do *Supabase Query*, foi adicionada a esse *container* a tabela referente aos usuários, aplicando um filtro em que o ID do usuário seja igual ao ID do usuário autenticado no banco de dados, isto é, logado no aplicativo (Figura 36). A mesma lógica foi aplicada ao *container* “Cadastro” (Figura 37).

Figura 36 - Filtro dos dados para o usuário logado



Fonte: Elaborado pelo autor.

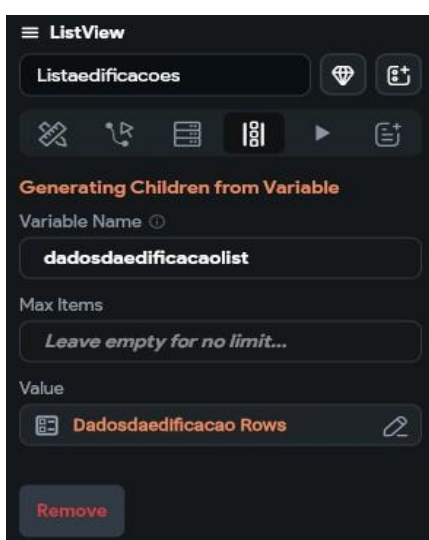
Figura 37 - Filtro dos dados da edificação para usuário logado



Fonte: Elaborado pelo autor.

Em relação às edificações inseridas, utilizou-se o *Generate Dynamic Children* para que as informações salvas no banco de dados fossem exibidas de forma dinâmica na tela do aplicativo (Figura 38). Dessa forma, vinculou-se as informações a serem exibidas na “*Listview Listaedificacoes*” às linhas da tabela que trata dos dados da edificação no Supabase. Para lidar com um grande volume de edificações cadastradas, foi adicionado à estrutura da tela um campo de busca como mostrado na Figura 35.

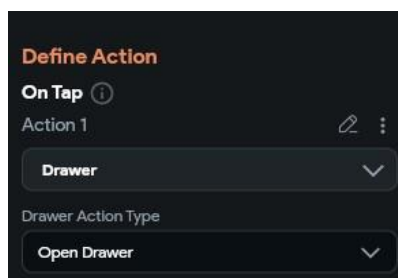
Figura 38 - Lista dos dados da edificação declarada no *Generate Dynamic Children*



Fonte: Elaborado pelo autor.

Algumas ações de acesso a outras funcionalidades foram definidas para exibir o menu de navegação e os componentes de salvar e editar a edificação cadastrada. Aos botões de Adicionar e Editar foram atribuídos comandos de abrir os componentes criados para cadastro e edição dos dados da edificação, respectivamente. O botão Menu abre o *Drawer* “*Menudenavegacao*” (Figura 35), que se trata do menu lateral, onde tem-se acesso ao botão Conta do usuário, que exibirá o componente Meu perfil com as informações do usuário, e ao botão Ajuda, que direciona à tela com as instruções iniciais sobre o aplicativo (Figura 44). As figuras a seguir ilustram as ações definidas para cada botão.

Figura 39 - Ação atribuída ao botão Menu



Fonte: Elaborado pelo autor.

Figura 40 - Ação atribuída ao botão Conta do usuário presente no Menu



Fonte: Elaborado pelo autor.

Figura 41 - Ação atribuída ao botão Ajuda presente no Menu

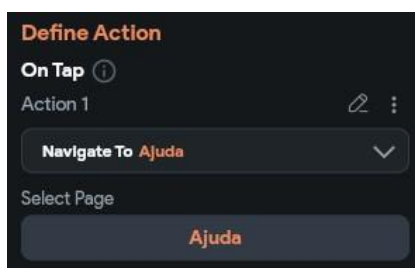
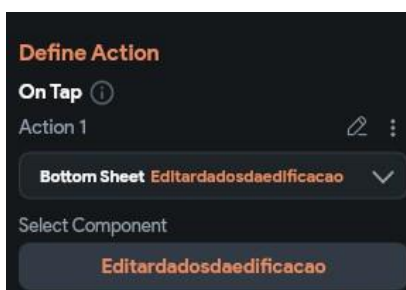
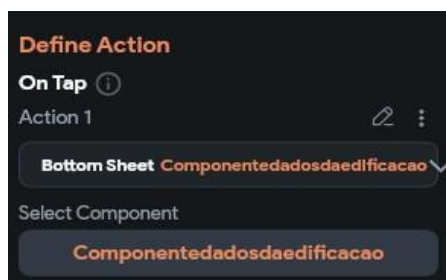


Figura 42 - Ação atribuída ao botão Editar



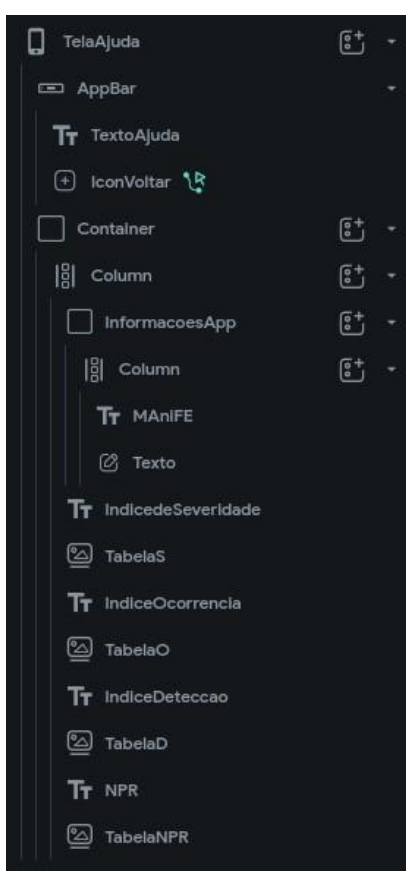
Fonte: Elaborado pelo autor.

Figura 43 - Ação atribuída ao botão Adicionar



Fonte: Elaborado pelo autor.

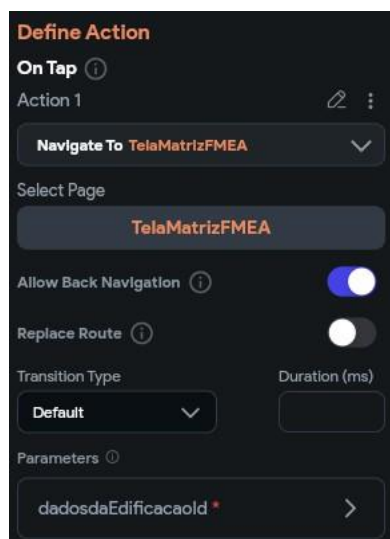
Figura 44 - Estrutura da tela Ajuda



Fonte: Elaborado pelo autor.

O botão de Acesso, apresentado na Figura 35, direciona o usuário à tela Matriz FMEA (que será apresentada na próxima seção), através da definição de um parâmetro que permite a visualização das manifestações patológicas cadastradas em cada edificação.

Figura 45 - Ação do botão de acesso à tela Matriz FMEA

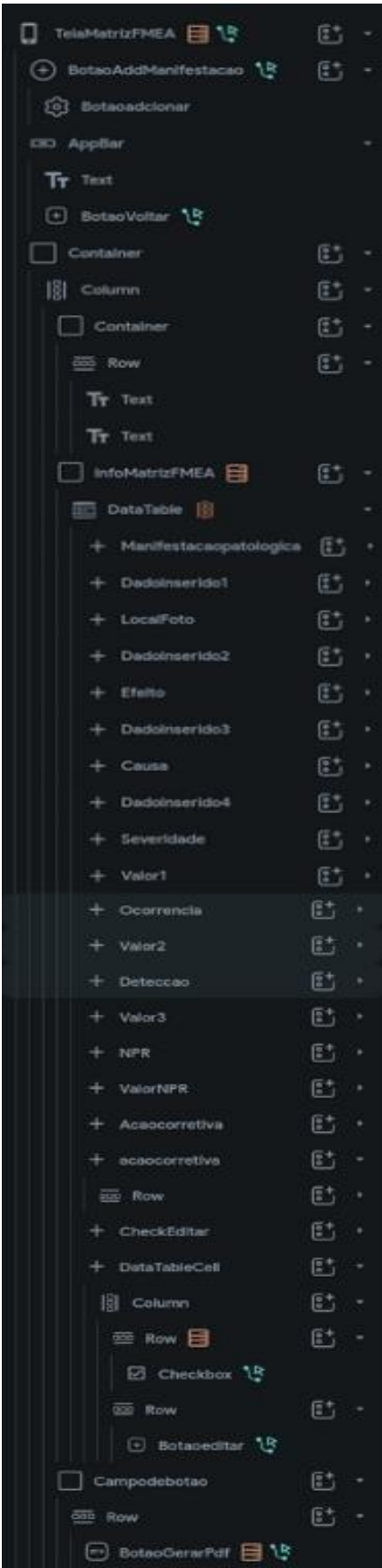


Fonte: Elaborado pelo autor.

3.4.4.4 Tela Matriz FMEA

Nesta tela, o usuário poderá cadastrar as manifestações patológicas encontradas na edificação onde se está realizando a inspeção predial, bem como editar os dados inseridos. As manifestações patológicas serão exibidas em ordem decrescente de valor de NPR, índice que será calculado automaticamente no aplicativo com base nos valores de Severidade, Ocorrência e Detecção. Outra funcionalidade disponível nesta tela é a ação de gerar um PDF, que reunirá todas as informações relacionadas à edificação e às manifestações patológicas.

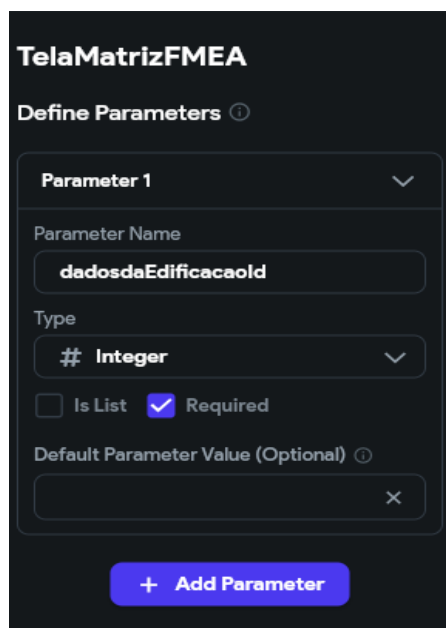
Figura 46 - Estrutura da tela Matriz FMEA



Fonte: Elaborado pelo autor.

Para garantir que as informações corretas sejam exibidas para a edificação selecionada, foi necessário criar um parâmetro para a tela (Figura 47). Esse parâmetro foi utilizado para filtrar as manifestações patológicas de acordo com ID da edificação, conforme a Figura 48.

Figura 47 - Parâmetro para tela Matriz FMEA



TelaMatrizFMEA

Define Parameters ⓘ

Parameter 1 ▾

Parameter Name

dadosdaEdificacaoId

Type

Integer ▾

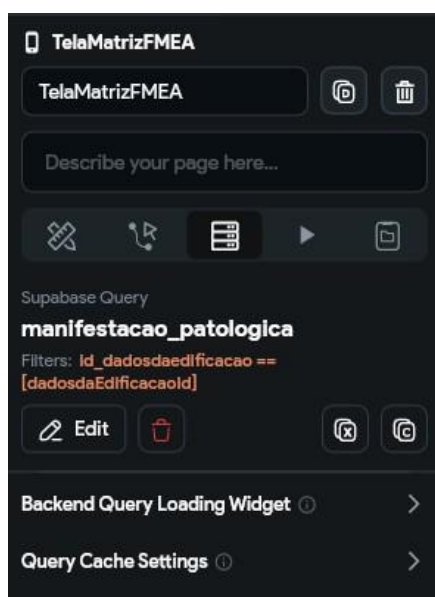
☐ Is List ☒ Required

Default Parameter Value (Optional) ⓘ

+ Add Parameter

Fonte: Elaborado pelo autor.

Figura 48 - Filtro na tabela de manifestações patológicas conforme o ID da edificação



TelaMatrizFMEA

TelaMatrizFMEA ⓘ ⌵

Describe your page here...

Supabase Query

manifestacao_patologica

Filters: **id_dadosdaedificacao ==**
[dadosdaEdificacaoId]

Edit **Delete** **Copy** **Paste**

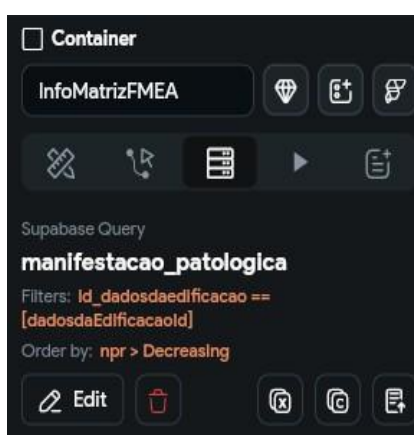
Backend Query Loading Widget ⓘ >

Query Cache Settings ⓘ >

Fonte: Elaborado pelo autor.

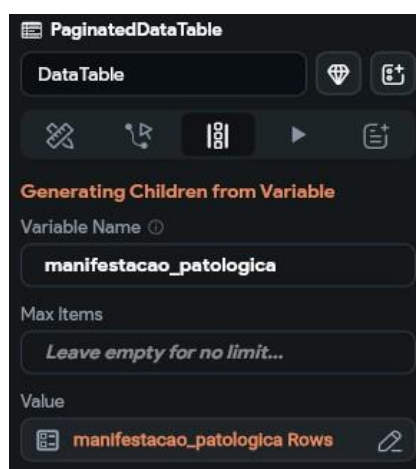
O *container* “InfoMatrizFMEA”, apresentado na Figura 46, contém um vínculo com a tabela onde serão inseridas as informações. Assim, esse componente foi integrado ao banco de dados que armazena as manifestações patológicas, estabelecendo o mesmo filtro aplicado anteriormente. Para esse *container*, foi determinada, através do recurso *Order by*, a organização em ordem decrescente sob o critério dos valores do NPR. Além disso, para que os valores fossem exibidos de forma dinâmica utilizou-se o *Generate Dynamic Children* com um vínculo entre as linhas de dados inseridos na estrutura de tabela chamada “*DataTable*”, do aplicativo, e as linhas da tabela “Dadosmanifestacaopatologica”, do Supabase.

Figura 49 - Vínculo e filtro aplicado ao *container* “InfoMatrizFMEA”



Fonte: Elaborado pelo autor.

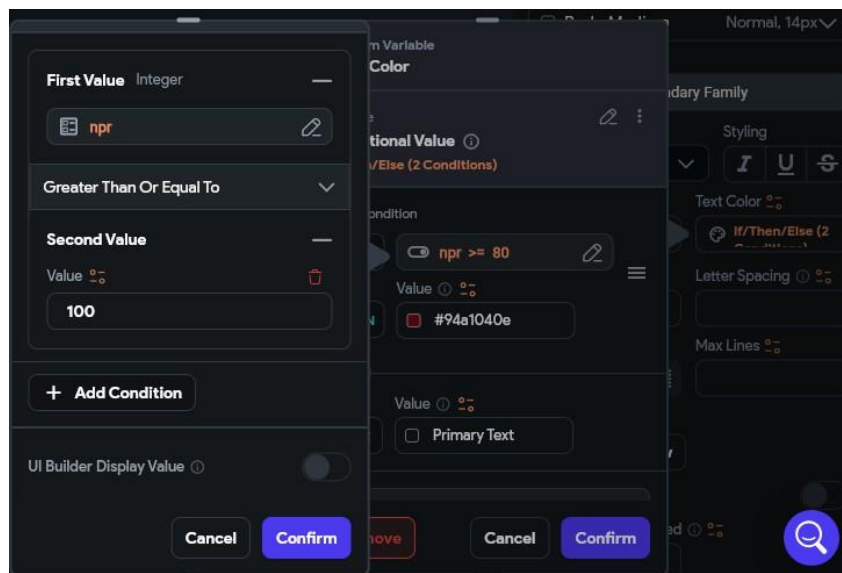
Figura 50 – “*DataTable*” vinculada às linhas da tabela do banco de dados



Fonte: Elaborado pelo autor.

Na coluna NPR da “*DataTable*”, foi determinada uma condicionante para que o valor seja exibido na cor vermelha quando os valores forem iguais ou superiores a 100, indicando graus de riscos altos, segundo a revisão de literatura.

Figura 51 - Condicionante de exibição dos valores de NPR na tabela “*DataTable*”



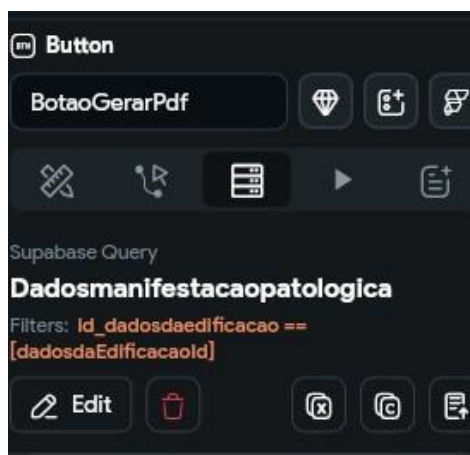
Fonte: Elaborado pelo autor.

Para os botões dessa tela, foram adicionados comandos para voltar à tela de início, adicionar, editar informações e gerar o PDF. O botão de adicionar e editar abrirão os componentes de cadastro e edição dos dados das manifestações patológicas, descritos na seção 3.4.4.2.

Além dessas funcionalidades, o “*Checkbox*”, apresentado na Figura 46, que tem a função de checar se a ação corretiva foi efetiva, utiliza a ação *BackendCall: Update Rows*. Se o usuário selecionar a caixa, o valor *True* será exibido na coluna *Check* no Supabase. Caso contrário, será exibido *False*.

Ao botão Gerar PDF, vinculou-se a tabela “*Dadosmanifestacaopatologica*”, estabelecendo o filtro com a relação de igualdade entre o ID da tabela referente aos dados da edificação e o parâmetro de tela (Figura 52). Assim, ao gerar o documento PDF será mostrada as informações da edificação à qual estão sendo registradas as manifestações patológicas.

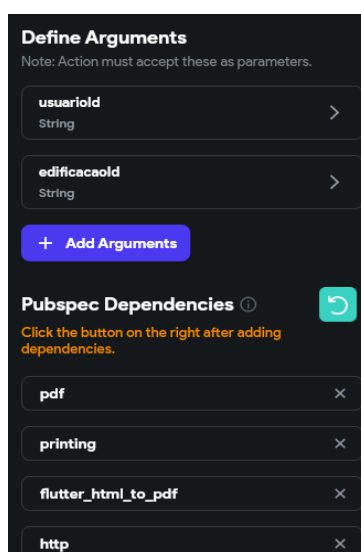
Figura 52 - Botão de gerar PDF com vínculo e filtro



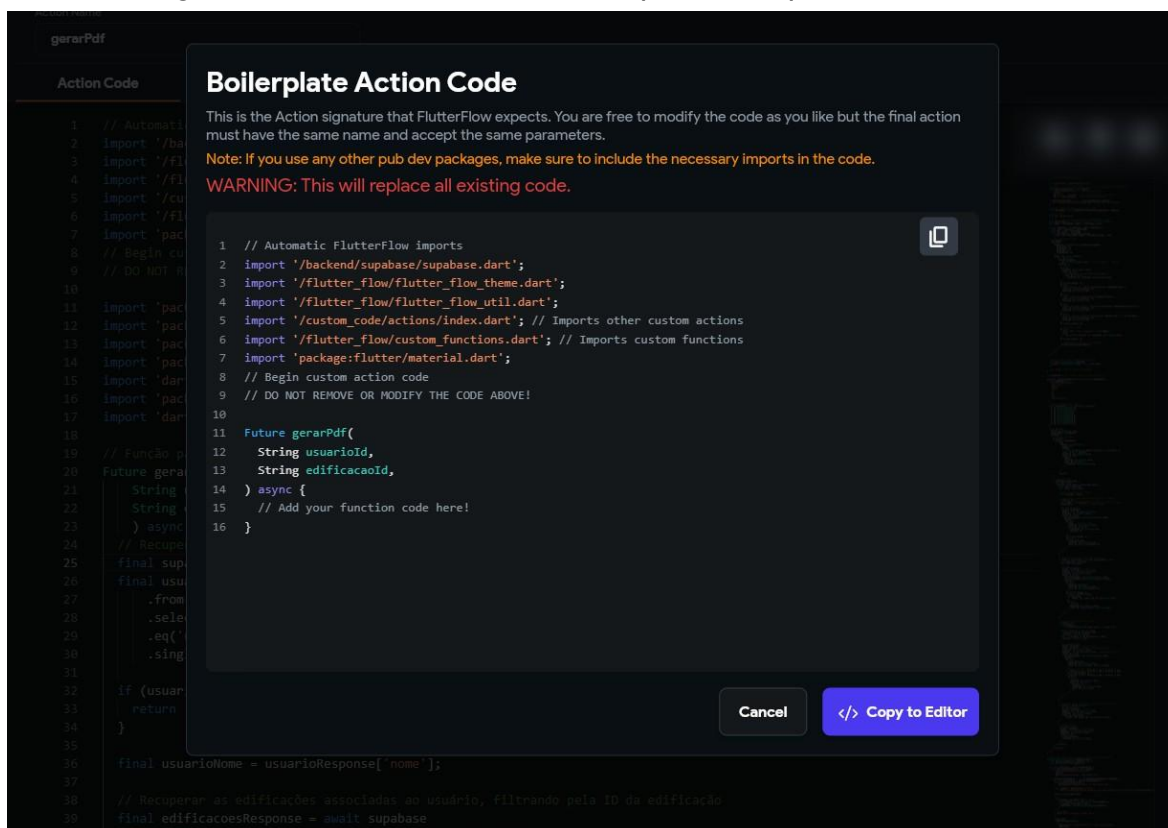
Fonte: Elaborado pelo autor.

Para possibilitar a emissão de um documento dentro do aplicativo, utilizou-se o recurso *Custom Action* do *FlutterFlow*, que permite ao desenvolvedor customizar ações utilizando código em Dart. Nesta funcionalidade, é declarado os argumentos que serão usados para criação da ação, que são o “usuarioid” e a “edificacaoid”, ambos do tipo *string*. Isso é importante para que a ação de gerar o documento contenha as informações referente ao usuário e à edificação. Em seguida, é definido as dependências necessárias (Figura 53) para a ação customizada. Após isso, por meio do *BoilerPlate Action Code*, obteve-se o bloco básico para estruturação do código, como mostrado na Figura 54.

Figura 53 - Definição dos argumentos e dependências



Fonte: Elaborado pelo autor.

Figura 54 - Bloco básico fornecido pelo *Boilerplate Action Code*

Fonte: Elaborado pelo autor.

Além das importações de bibliotecas padrões do *FlutterFlow*, outras foram necessárias, conforme ilustrado a seguir com cada função descrita.

Figura 55 - Bibliotecas adicionais

```

import 'package:pdf/pdf.dart'; // Base para criar o pdf
import 'package:pdf/widgets.dart' as pw; // Adicionar widgets no pdf
import 'package:printing/printing.dart'; // Renderizar, imprimir e compartilhar pdf
import 'package:http/http.dart' as http; // Usado para carregar imagens via URL
import 'dart:typed_data'; // Usado para trabalhar com imagens em bytes
import 'package:image_picker/image_picker.dart'; // Usado para selecionar imagem da galeria
import 'dart:io'; // Usado para abrir imagens

```

Fonte: Elaborado pelo autor.

Inicialmente, o código para gerar o PDF verifica o usuário e a edificação associada para a recuperação dos dados que serão mostrados no documento. O conteúdo, então, é dividido em páginas geradas dinamicamente, em que o número total de páginas é calculado com base na quantidade de dados a serem incluídos. Foi estabelecido para cada página duas linhas da tabela, conforme a Figura 56.

Figura 56 - Código para verificação das informações

```

19 // Função para gerar o PDF
20 Future gerarPdf(
21     String usuarioId, // Identificador do usuário logado
22     String edificacaoId // ID da edificação a ser filtrada
23 ) async {
24     // Para recuperar o usuário logado no aplicativo
25     final supabase = Supabase.instance.client;
26     final usuarioResponse = await supabase
27         .from('Usuario') // Tabela do usuário
28         .select()
29         .eq('user_id',
30             usuarioId) // Filtra pelo ID do usuário logado no aplicativo
31         .single();
32
33     if (usuarioResponse == null) {
34         return 'Usuário não encontrado';
35     }
36
37     final usuarioNome = usuarioResponse['nome'];
38
39     // Para recuperar as edificações associadas ao usuário, filtrando pela ID da edificação
40     final edicoesResponse = await supabase
41         .from('Dadosdaedificacao') // Tabela das edificações
42         .select()
43         .eq('id_user', usuarioId) // Filtra pela associação com o usuário
44         .eq('id', edificacaoId); // Filtro pela ID da edificação
45
46     if (edicoesResponse == null || edicoesResponse.isEmpty) {
47         return 'Nenhuma edificação encontrada para o usuário com o ID informado';
48     } // Verificação para garantir que há dados válidos relacionada a edificações de um usuário
49
50     // Para buscar os dados da tabela antes de criar o documento PDF
51     final tabelaDados = await _getDataForTable(edicoesResponse, supabase);
52
53     // Gerar o PDF
54     final pdf = pw.Document();
55
56     // Dividir os dados da tabela em blocos para a página do documento PDF
57     const rowsPerPage = 2; // Número de linhas para uma página
58     final pages = (tabelaDados.length / rowsPerPage)
59         .ceil(); // Adiciona páginas conforme os dados na página
60
61     for (int pageIndex = 0; pageIndex < pages; pageIndex++) {
62         // Índice da página atual
63         final start = pageIndex * rowsPerPage;
64         final end = start + rowsPerPage;
65         final tabelaPagina = tabelaDados.sublist(
66             start, end > tabelaDados.length ? tabelaDados.length : end);
67     }

```

Fonte: Elaborado pelo autor.

Posteriormente, foi determinado o formato do PDF em modo paisagem, onde em cada página será exibido um cabeçalho contendo informações da edificação inseridas pelo usuário no aplicativo, que serão extraídas do banco de dados através de uma consulta às colunas específicas. O PDF será gerado dinamicamente em páginas sucessivas, sendo que cada conterà o cabeçalho e uma parte da tabela correspondente.

Figura 57 - Código para organização do layout e cabeçalho do PDF

```

68 // Adicionar página ao PDF e estruturação o layout da pagina com a definicao da orientação da página e margens
69 pdf.addPage(
70   pw.Page(
71     pageFormat: PdfPageFormat.a4.landscape.copyWith(
72       marginLeft: 10,
73       marginRight: 10,
74       marginTop: 15,
75       marginBottom: 10,
76     ),
77     build: (pw.Context context) {
78       return pw.Column(
79         crossAxisAlignment: pw.CrossAxisAlignment
80         .start, // Alinha as informações à esquerda da página
81         children: [
82           // Cabeçalho
83           pw.Align(
84             alignment: pw.Alignment
85             .center, // Alinha as informações ao centro da página
86             child: pw.Text(
87               'Hierarquização das manifestações patológicas', // Título a ser alinhado
88               style: pw.TextStyle(
89                 fontSize: 14,
90                 fontWeight: pw.FontWeight.bold), // Estilo do Título
91             ),
92           ),
93           pw.SizedBox(height: 10),
94           pw.Text(
95             'Nome da Edificação: ${edificacoesResponse[0]['nomedaedificacao']}', // Texto "Nome da edificação"
96             style: pw.TextStyle(fontSize: 12), // Estilo do texto
97             textAlign: pw.TextAlign.left,
98           ),
99           pw.Text(
100             'Local: ${edificacoesResponse[0]['localdaedificacao']}', // Texto "Local"
101             style: pw.TextStyle(fontSize: 12), // Estilo do texto
102             textAlign: pw.TextAlign.left,
103           ),
104           pw.Text(
105             'Responsável pela vistoria: ${edificacoesResponse[0]['nomedoresponsaveltecnico']}', // Texto "Responsável técnico"
106             style: pw.TextStyle(fontSize: 12), // Estilo do texto
107             textAlign: pw.TextAlign.left,
108           ),
109           pw.Text(
110             'Data da vistoria: ${edificacoesResponse[0]['datadavistoria']}', // Texto "Data da vistoria"
111             style: pw.TextStyle(fontSize: 12), // Estilo do texto
112             textAlign: pw.TextAlign.left,
113           ),
114           pw.SizedBox(height: 20),
115           // Título Matriz FMEA
116           pw.Text(
117             'Matriz FMEA - Página ${pageIndex + 1} de $pages', //Texto "Matriz FMEA" com a paginação
118             style: pw.TextStyle(
119               fontSize: 12,
120               fontWeight: pw.FontWeight.bold), // Estilo do texto
121           ),
122           pw.SizedBox(height: 20),
123           // Construir a tabela personalizada para a página atual conforme as informações do banco de dados
124           _buildCustomTable(tabelaPagina),
125         ],
126       );
127     },
128   );
129 },
130 );
131 );
132 }
133
134 // Salvar o PDF ou gerar o arquivo para download
135 final file = await Printing.layoutPdf(
136   onLayout: (PdfPageFormat format) async => pdf.save(),
137 );
138
139 return file; // Retorna o arquivo PDF gerado
140 }

```

Fonte: Elaborado pelo autor.

Em seguida foi elaborado o código para a criação da tabela customizada para a exibição das informações das manifestações patológicas. A tabela contém cabeçalhos com os mesmos nomes da tabela do banco de dados e larguras das colunas definidas pela função “*columnWidths*”. Na primeira coluna, adicionou-se um contador (“*ordemGlobal*”) para numeração da manifestação patológica. Já na coluna referente ao local e foto, é utilizada a estrutura condicional *If* e *Else* com o objetivo de verificar se existe uma foto disponível para exibição, caso contrário é retornado o texto “Sem foto”. Na coluna NPR, também é realizada uma análise condicional, na qual dependendo do valor, a célula será preenchida com uma cor previamente estabelecida. Além disso, quando o valor desse índice for igual a 50, 100 e 200, será exibido o texto “Análise”, orientando o responsável pela inspeção a avaliar o grau de risco em que a manifestação patológica se encaixa. O código apresentado nas figuras a seguir descreve essas configurações de forma detalhada.

Figura 58 - Código para a organização do *layout* da tabela

```

142 // Layout da tabela que terá as manifestações patológicas
143
144 int ordemGlobal =
145 | 1; // Contador global para ser exibido na coluna das manifestações
146
147 pw.Widget_buildCustomTable(List<List<dynamic>> tabelaDados) {
148 //Cada item da tabelaDados representa uma linha da tabela,
149 const tableHeaders = [
150 //e cada linha é uma lista de valores
151 'Manifestação Patológica',
152 'Local/Foto',
153 'Efeito',
154 'Causa', // Cabeçalhos da tabela
155 'S',
156 'O',
157 'D',
158 'NPR',
159 'Ação corretiva',
160 ];
161
162 // Definição da largura das colunas
163 final columnWidths = <int, pw.TableColumnWidth>{
164 0: pw.FlexColumnWidth(1.5),
165 1: pw.FlexColumnWidth(2.5),
166 2: pw.FlexColumnWidth(2),
167 3: pw.FlexColumnWidth(2),
168 4: pw.FlexColumnWidth(0.5),
169 5: pw.FlexColumnWidth(0.5),
170 6: pw.FlexColumnWidth(0.5),
171 7: pw.FlexColumnWidth(0.8),
172 8: pw.FlexColumnWidth(1.7),
173 };
174
175 return pw.Table(
176 /// Cria a tabela no pdf
177 border: pw.TableBorder.all(),
178 columnWidths: columnWidths,
179 children: [

```

Fonte: Elaborado pelo autor.

Figura 59 - Continuação do código para a organização do *layout* da tabela

```

179     children: [
180         // Estilo do cabeçalho da tabela
181         pw.TableRow(
182             children: tableHeaders
183             .map(
184                 (header) => pw.Container(
185                     height: 35, //Altura do cabeçalho
186                     padding: const pw.EdgeInsets.symmetric(
187                         vertical: 4,
188                         horizontal: 8), // Espaço do texto dentro da célula
189                     color: PdfColors.grey300, // Cor de fundo
190                     alignment: pw.Alignment.center,
191                     child: pw.Text(
192                         header, // Alinhamento do texto ao centro da célula
193                         textAlign: pw.TextAlign.center, // Estilo negrito do texto
194                         style: pw.TextStyle(
195                             fontWeight: pw.FontWeight.bold,
196                         ),
197                     ),
198                 ),
199             )
200         ).toList(), //Envio para alimentação dos dados
201     ),
202
203     // Linhas da tabela
204
205     ...tabelaDados.map((linha) {
206         ///percorre cada linha da tabela
207         return pw.TableRow(
208             children: linha.asMap().entries.map((entry) {
209                 final index = entry.key; //Índice da célula
210                 final celula = entry.value; //Valor da célula
211
212                 // Definir altura fixa das linhas
213                 final cellHeight = 195.0;
214
215                 // Lógica para a coluna "Manifestação Patológica" (coluna 1 - índice 0)
216                 if (index == 0) {
217                     final ordem =
218                         '${ordemGlobal}°'; // Adiciona o número de ordem usando o conatdor global já declarado
219                     ordemGlobal++; // Incrementa o contador global
220
221                     // Criação da célula
222                     return pw.Container(
223                         height: cellHeight,
224                         padding: const pw.EdgeInsets.all(4),
225                         alignment: pw.Alignment.center,
226                         child: pw.Column(
227                             mainAxisAlignment: pw.MainAxisAlignment.center,
228                             children: [
229                                 pw.Text(
230                                     ordem, // Ordem (1º,2º...)
231                                     textAlign: pw.TextAlign.center,
232                                     style: pw.TextStyle(
233                                         // Estilo do texto
234                                         fontWeight: pw.FontWeight.bold,
235                                         fontSize: 14,
236                                     ),
237                                 ),
238                                 pw.SizedBox(height: 4),
239                                 pw.Text(
240                                     celula.toString(),
241                                     textAlign: pw.TextAlign.center,
242                                     style: pw.TextStyle(fontSize: 12), // Estilo do texto
243                                 ),
244                             ],
245                         ),
246                     );
247                 }
248
249                 // Lógica para a coluna "Local/Foto" (coluna 2 - índice 1)
250                 if (index == 1 && celula is List && celula.length == 2) {
251                     final nomeLocal = celula[0];
252                     final foto = celula[1];
253

```

Fonte: Elaborado pelo autor.

Figura 60 - Continuação do código para a organização do *layout* da tabela

```

254 return pw.Container(
255     height: cellHeight, // Altura fixa (195)
256     padding: const pw.EdgeInsets.all(4),
257     alignment: pw.Alignment.center,
258     child: pw.Column(
259         mainAxisAlignment: pw.MainAxisAlignment
260         .center, // Centraliza verticalmente o texto
261         crossAxisAlignment: pw.CrossAxisAlignment
262         .center, // Centraliza horizontalmente o texto
263         children: [
264             pw.Text(
265                 nomeLocal.toString(),
266                 textAlign: pw.TextAlign.center,
267                 style: pw.TextStyle(fontSize: 12),
268             ),
269             pw.SizedBox(height: 8),
270             if (foto is pw.MemoryImage)
271                 pw.Image(foto,
272                     height: 135,
273                     width: 135,
274                     fit: pw.BoxFit
275                     .cover) // Tamanho da foto caso seja anexada
276             else
277                 pw.Text(
278                     'Sem foto', // Texto exibido caso não seja anexada a foto
279                     textAlign: pw.TextAlign.center,
280                     style: pw.TextStyle(color: PdfColors.grey),
281                 ),
282         ],
283     ),
284 );
285 }
286
287 // Lógica para a coluna "NPR" (coluna 8 - índice 7)
288 if (index == 7) {
289     final nprValue =
290         int.tryParse(celula.toString().split(' ').first) ?? 0;
291
292     // Condicional para cor da célula
293     PdfColor cellColor = PdfColors.white;
294     if (nprValue >= 50 && nprValue <= 99) {
295         // Limites para exibição da célula na cor amarela
296         cellColor = PdfColors.yellow;
297     } else if (nprValue >= 100 && nprValue <= 1000) {
298         // Limites para exibição da célula na cor vermelha
299         cellColor = PdfColors.red;
300     }
301
302     return pw.Container(
303         height: cellHeight, // Altura fixa (195)
304         color: cellColor, // Define a cor de preenchimento
305         alignment: pw.Alignment.center,
306         padding: const pw.EdgeInsets.all(4), // Espaçamento interno
307         child: pw.Column(
308             mainAxisAlignment: pw.MainAxisAlignment
309             .center, // Centraliza o valor verticalmente
310             children: [
311                 pw.Text(
312                     celula.toString(),
313                     textAlign: pw.TextAlign.center,
314                     style:
315                         pw.TextStyle(color: PdfColors.black), // Cor da fonte
316                 ),
317                 // Mostrar "Análise" apenas para os valores específicos
318                 if (nprValue == 50 ||
319                     nprValue == 100 ||
320                     nprValue ==
321                     200) // Valores =50 ou 100 ou 200, exibe "Análise"
322                     pw.SizedBox(
323                         height:
324                         4), // Espaço vertical para afastar o valor e texto
325                 if (nprValue == 50 || nprValue == 100 || nprValue == 200)
326                     pw.Text(
327                         'Análise',
328                         textAlign:
329                             pw.TextAlign.center, // Estilo do texto a ser exibido

```

Fonte: Elaborado pelo autor.

Figura 61 - Continuação do código para a organização do *layout* da tabela

```

329         pw.TextAlign.center, //Estilo do texto a ser exibido
330         style: pw.TextStyle(
331             fontSize: 12,
332             fontWeight: pw.FontWeight.bold,
333             color: PdfColors.black,
334         ),
335     ),
336 ],
337 ),
338 );
339 }
340
341 // Renderizar imagem diretamente (caso não seja "Local/Foto")
342 if (celula is pw.MemoryImage) {
343     //Verifica se há imagem
344     return pw.Container(
345         height: cellHeight, // Altura fixa
346         padding: const pw.EdgeInsets.all(1),
347         alignment: pw.Alignment.center,
348         child: pw.Image(celula,
349             height: 400,
350             width: 400,
351             fit: pw.BoxFit.cover), // Imagem é renderizada com 400px
352     );
353 }
354
355 // Células normais, caso não seja com foto
356 return pw.Container(
357     height: cellHeight, // Altura fixa (195)
358     padding: const pw.EdgeInsets.all(4),
359     alignment: pw.Alignment.center,
360     child: pw.Text(
361         //Estilo do texto
362         celula.toString(),
363         textAlign: pw.TextAlign.center,
364     ),
365 );
366 );
367 }).toList(),
368 ],
369 );
370 );
371 }
372

```

Fonte: Elaborado pelo autor.

Para a busca no banco de dados, carregamento das imagens e organização das informações para exibição na tabela customizada, utilizou-se a função “*_getDataForTable*” que recebe dois parâmetros: a lista de edificações e a instância do Supabase. As manifestações patológicas são consultadas com base na edificação associada, utilizando a chave estrangeira “*id_dadosdaedificacao*”. As fotos anexadas no aplicativo e encaminhadas para o banco de dados são carregadas pela URL, que é validada pela requisição HTTP. Em seguida, as linhas com os dados são adicionadas no documento de acordo com o nome de cada coluna da tabela do Supabase. Após o processamento, a tabela é ordenada e organizada de forma decrescente a partir do valor do NPR e, por fim, é retornada.

Figura 62 - Código para busca e organização das informações

```

373 // Função para buscar dados da tabela e carregar fotos
374 Future<List<List<dynamic>>> _getDataForTable(
375     List edificacoesResponse, supabase) async {
376     //Traz Dados da edificações do banco de dados
377     final List<List<dynamic>> tabela = []; //Tabela criada para receber os dados
378
379     for (final edificacao in edificacoesResponse) {
380         final manifestacoesResponse = await supabase
381             .from('Dadosmanifestacaopatologica')
382             .select()
383             .eq(
384                 'id_dadosdaedificacao',
385                 edificacao[
386                     'id']); // Busca manifestações patológicas para cada edificação
387
388         if (manifestacoesResponse != null && manifestacoesResponse.isNotEmpty) {
389             for (final manifestacao in manifestacoesResponse) {
390                 String manifestacaopatologica = manifestacao['nomedapatologia'] ??
391                     'Não especificada'; //Mensagem caso manifestação não tenha sido cadastradas
392
393                 final imageUrl = manifestacao['Foto']; //URL da imagem no banco de dados
394                 ;
395
396                 pw.MemoryImage? imgProvider; //carregamento da foto
397
398                 try {
399                     if (imageUrl.startsWith('https://')) {
400                         //Validação da URL com https
401                         final response = await http.get(Uri.parse(imageUrl));
402                         if (response.statusCode == 200) {
403                             imgProvider = pw.MemoryImage(response.bodyBytes);
404                         }
405                     }
406                 } catch (e) {
407                     print(
408                         'Erro ao carregar a imagem: $e'); //Para caso erro no carregamento da foto
409                 }
410
411                 tabela.add([
412                     //Adição dos dados coletados no Supabase para a tabela
413                     manifestacaopatologica,
414                     [
415                         manifestacao['localdapatologia'] ?? 'Local não informado',
416                         imgProvider
417                     ],
418                     manifestacao['efeito'] ?? 'Efeito não informado',
419                     manifestacao['causa'] ?? 'Causa não informada',
420                     manifestacao['s'].toString(),
421                     manifestacao['o'].toString(),
422                     manifestacao['d'].toString(),
423                     manifestacao['npr'].toString(),
424                     manifestacao['acaocorretiva'].toString(),
425                 ]);
426             }
427         }
428     }
429
430     // Ordenando a tabela pelo valor NPR de forma decrescente
431     tabela.sort((a, b) {
432         final nprA = int.tryParse(a[7].toString()) ??
433             0; //Coversão paara inteiro da coluna do NPR
434         final nprB = int.tryParse(b[7].toString()) ?? 0;
435         return nprB.compareTo(nprA); // Comparação para ordenar de forma decrescente
436     });
437

```

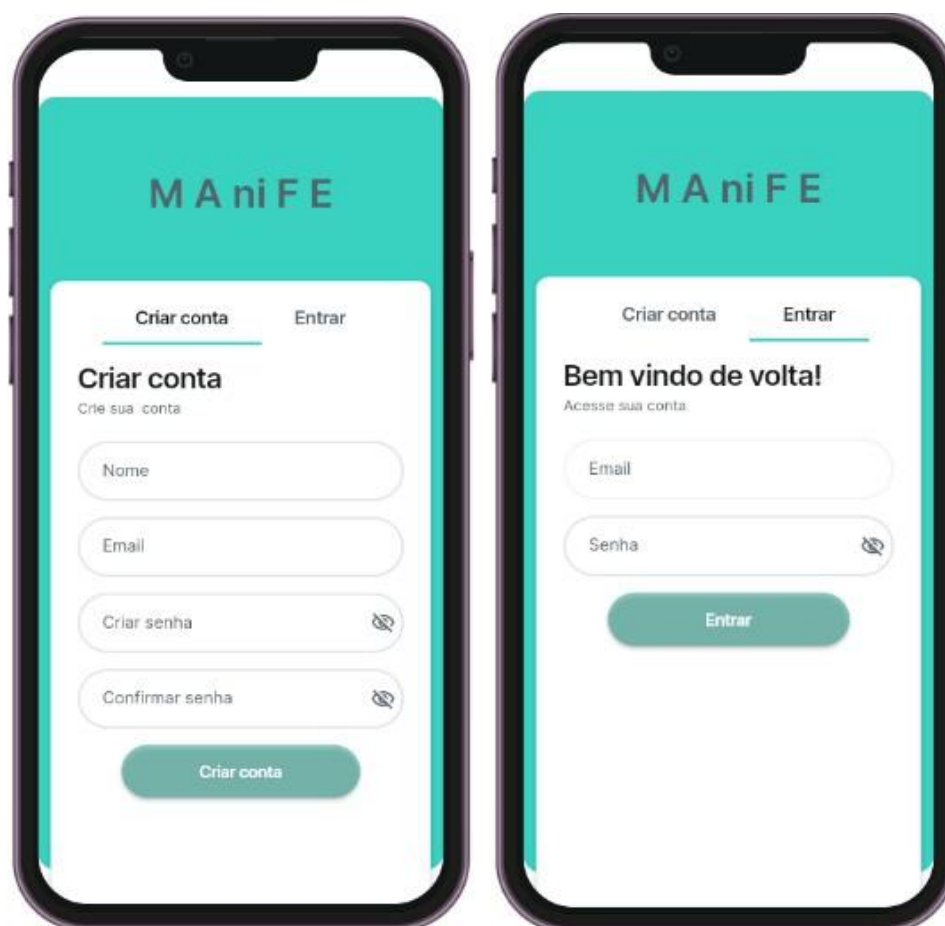
Fonte: Elaborado pelo autor.

4 RESULTADOS

4.1 INTERFACES GRÁFICAS DO APLICATIVO

O aplicativo inicia com a tela de autenticação do usuário, onde deverá ser preenchido os campos do formulário com as informações solicitadas. Nesta tela, o usuário pode acessar sua conta, caso já esteja cadastrada, ou realizar um cadastro para criar uma conta.

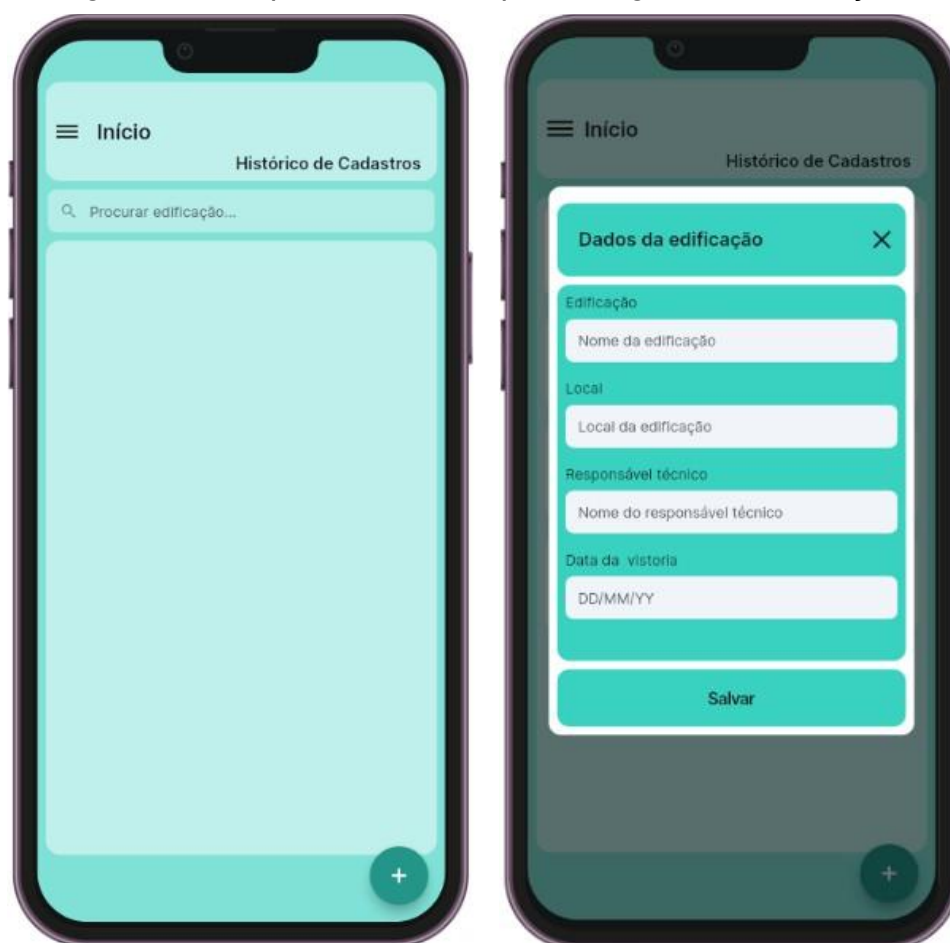
Figura 63 - Telas Entrar e Criar conta



Fonte: Elaborado pelo autor.

Após a autenticação, o usuário é direcionado para a tela de Início, onde terá acesso a uma barra de pesquisa e aos botões Menu e Adicionar. Ao clicar no botão Adicionar, será aberta uma tela contendo o formulário para o preenchimento dos dados referentes à edificação. A Figura 64 ilustra essa sequência.

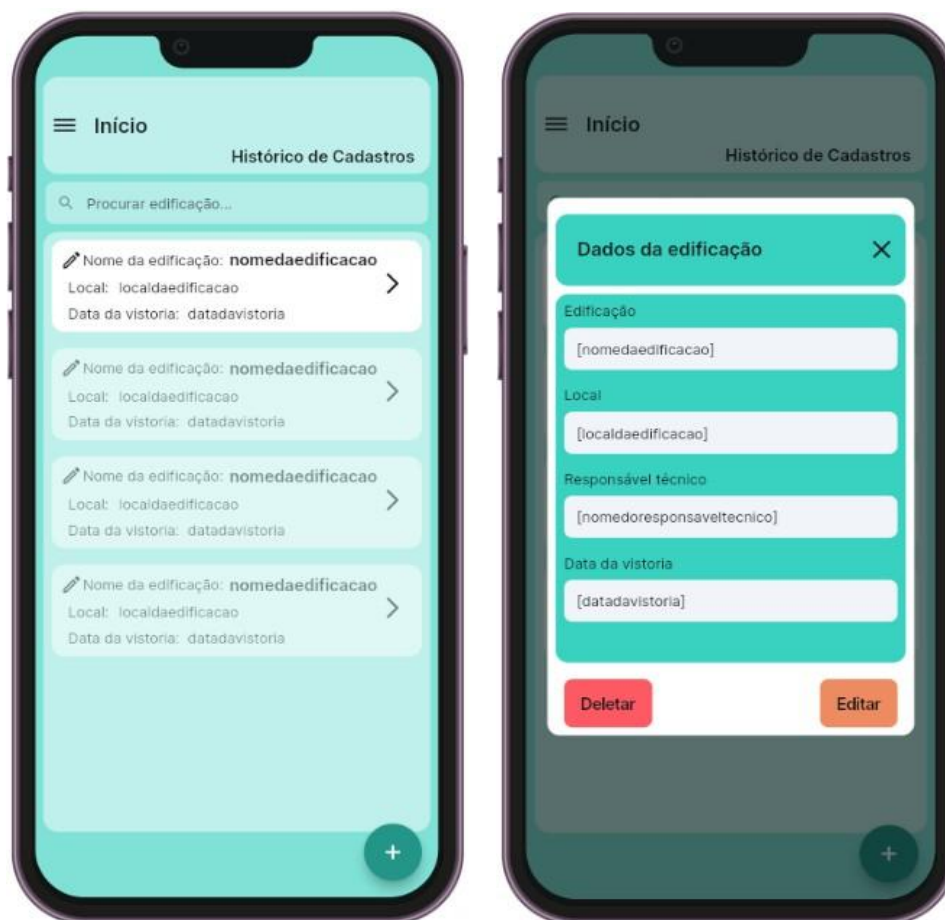
Figura 64 - Sequência de telas para o registro da edificação



Fonte: Elaborado pelo autor.

Ao salvar as informações do formulário, será possível visualizar um bloco com a edificação cadastrada, onde estarão localizados os botões para editar as informações e para acessar a tela da Matriz FMEA, apresentados seguir.

Figura 65 - Sequência de telas para edição dos dados



Fonte: Elaborado pelo autor.

Na tela da Matriz FMEA serão visualizados os botões de adicionar as manifestações patológicas e gerar o PDF, conforme a Figura 66.

Ao clicar no botão de adicionar, o usuário poderá inserir todas as informações observadas, bem como o registro fotográfico do problema. Essas informações serão exibidas na tela com o critério de NPR calculado e hierarquizado. Será possível ainda a edição das informações cadastradas, apresentada na Figura 67.

Por fim, o comando de gerar PDF possibilitará ao usuário criar um documento com todas as informações que foram inseridas no aplicativo, como ilustrado no exemplo da Figura 68. Nessa situação hipotética, é possível notar que no documento estão indicados os valores do NPR que necessitarão de análise para a definição do seu grau de risco, com base na revisão de literatura.

Nas Figuras 69 e 70 são apresentadas, respectivamente, as telas para sair da conta e as informações do aplicativo, acessadas por meio do menu na tela de Início.

Figura 66 - Sequência de telas para cadastrar as manifestações patológicas

The figure displays two sequential screens of a mobile application for FMEA (Failure Mode and Effects Analysis) matrix management.

Left Screen (Table View):

- Header: Matríz FMEA
- Field: Nome da Edificação: -
- Table with 2 columns: Manifestação patológica, Local/Foto
- Table content (4 rows):

Manifestação patológica	Local/Foto
-	-
-	-
-	-
-	-

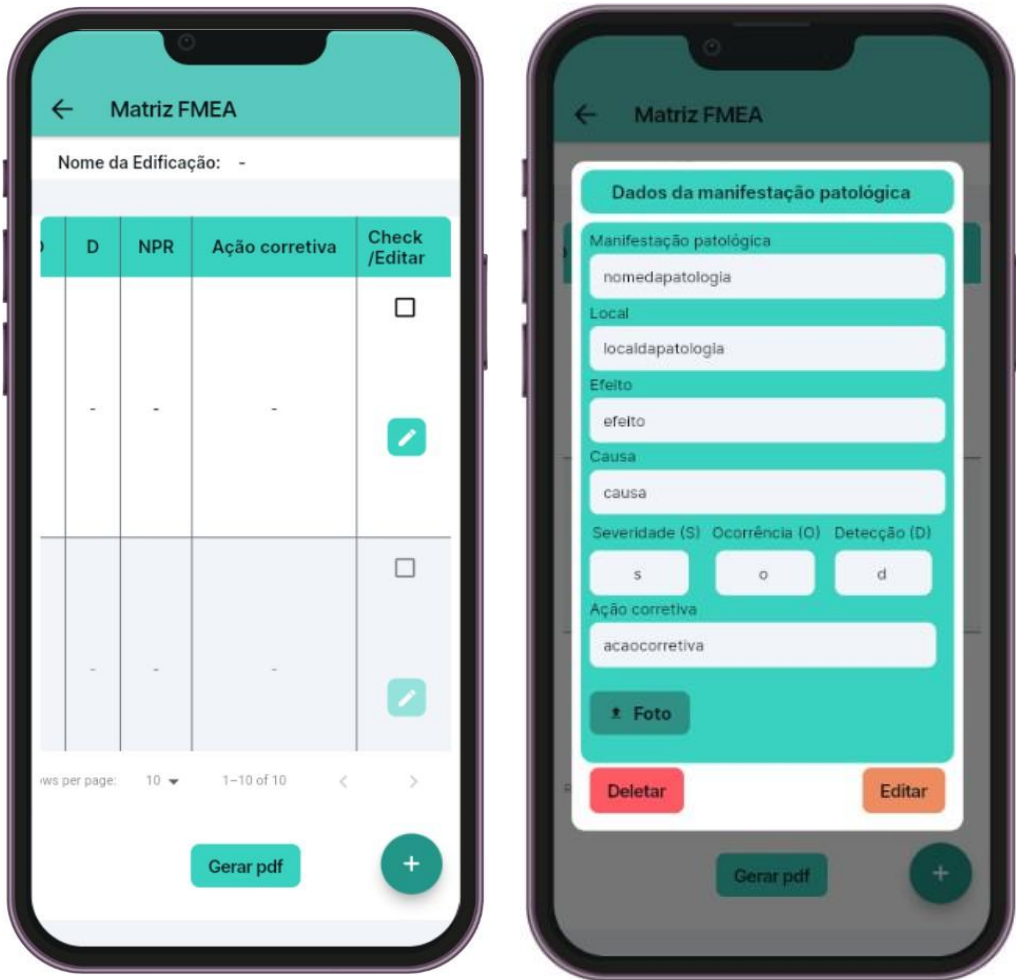
- Footer: Rows per page: 5, 1-4 of 4, Gerar pdf, +

Right Screen (Form View):

- Header: Matríz FMEA
- Section: Dados da manifestação patológica
- Fields: Manifestação patológica (Fissura, recalque...), Local (Local da manifestação patológica), Efeito (Dano estético, dano estrutural...), Causa (Infiltração, uso...)
- Fields: Severidade (S) (1 ao 10), Ocorrência (O) (1 ao 10), Detecção (D) (1 ao 10)
- Field: Ação corretiva (Tratamento da manifestação patológica)
- Buttons: Foto, Fechar, Salvar
- Footer: Gerar pdf, +

Fonte: Elaborado pelo autor.

Figura 67 - Sequência de telas para edição dos dados



Fonte: Elaborado pelo autor.

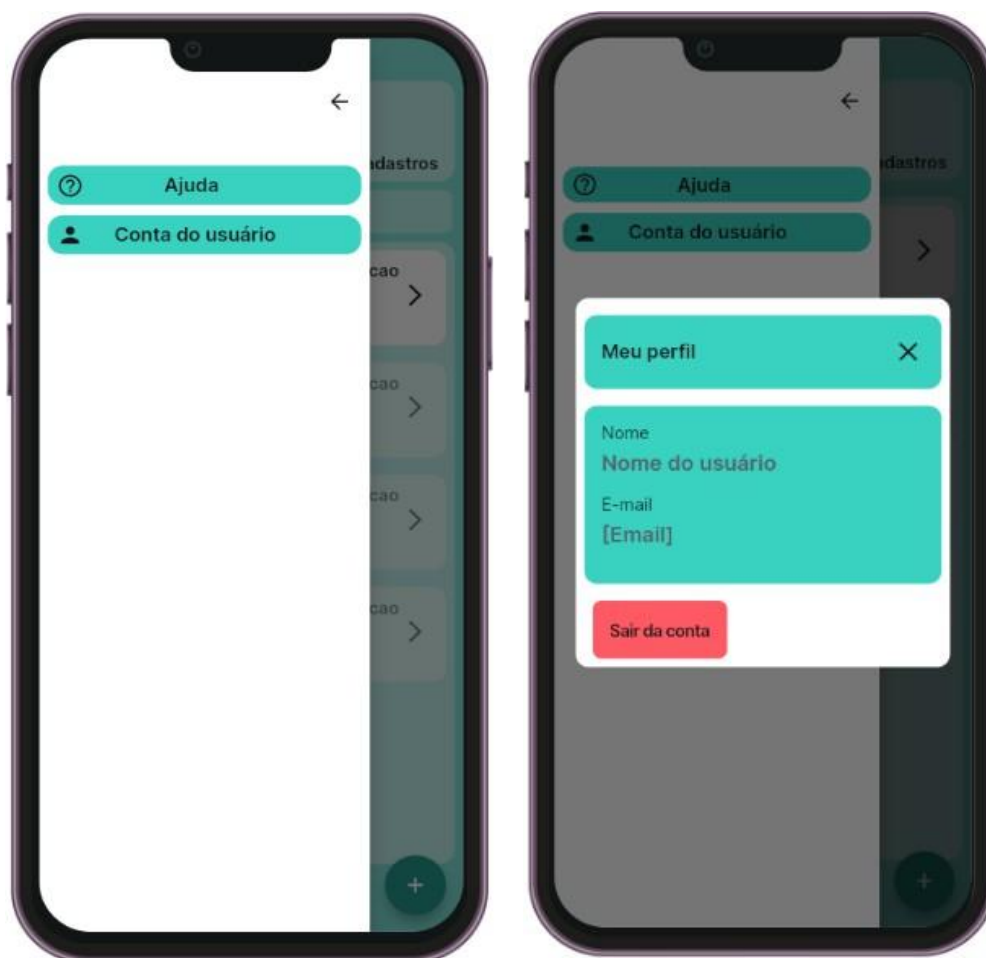
Figura 68 - Exemplo do documento PDF

Hierarquização das manifestações patológicas								
Nome da Edificação: nomedaedificacao Local: localdaedificacao Responsável pela vistoria: nomedoresponsaveltecnico Data da vistoria: datadavistoria								
Matriz FMEA - Página 1 de 2								
Manifestação Patológica	Local/Foto	Efeito	Causa	S	O	D	NPR	Ação corretiva
1ª Manifestação patológica 3	Local 3  Foto teste 3	Efeito 3	Causa 3	5	5	8	200 Análise	Ação corretiva 3
2ª Manifestação patológica 2	Local 2  Foto teste 2	Efeito 2	Causa 2	5	4	5	100 Análise	Ação corretiva 2

Hierarquização das manifestações patológicas								
Nome da Edificação: nomedaedificacao Local: localdaedificacao Responsável pela vistoria: nomedoresponsaveltecnico Data da vistoria: datadavistoria								
Matriz FMEA - Página 2 de 2								
Manifestação Patológica	Local/Foto	Efeito	Causa	S	O	D	NPR	Ação corretiva
3ª Manifestação patológica 1	Local 1  Foto teste 1	Efeito 1	Causa 1	2	5	5	50 Análise	Ação corretiva 1

Fonte: Elaborado pelo autor.

Figura 69 - Sequência de telas para sair da conta do aplicativo



Fonte: Elaborado pelo autor.

Figura 70 - Tela de Ajuda com as informações dos índices da FMEA



Fonte: Elaborado pelo autor.

4.2 TESTE DE FUNCIONALIDADE DO APLICATIVO

Nesta seção será apresentado o teste de funcionalidade do protótipo do aplicativo desenvolvido. Para isso, utilizou-se como base o estudo de caso do trabalho acadêmico de Emilyn Carolina Regos, que realizou uma análise das manifestações patológicas em uma edificação residencial unifamiliar em alvenaria convencional, utilizando a metodologia FMEA. O processo de inspeção foi de caráter sensorial, no qual foram identificadas as manifestações patológicas e suas quantidades para análise.

A autora elaborou uma tabela com o agrupamento das manifestações patológicas, especificando a quantidade de vezes em que cada tipo foi identificado. A presente seção tem como foco a tabela que utiliza a metodologia FMEA. Os resultados obtidos pelo estudo são apresentados em duas tabelas. Na primeira tabela, tem-se a estrutura do método, contendo apenas as informações preenchidas (Tabela 10). Na segunda tabela, é apresentado a hierarquização das manifestações patológicas de acordo com o índice do NPR, como mostrado na Tabela 11.

Tabela 10 - Aplicação da metodologia FMEA

Nº	Manifestação patológica	Efeito	Possíveis Causas	Fatores			NPR
				S	O	D	
1	Umidade accidental	Dano estético, abertura para novas anomalias.	Águas pluviais, falta de rufos.	4	3	3	36
2	Umidade ascensional	Dano estético, abertura para novas anomalias.	Falta de impermeabilização das vigas baldrames.	5	3	4	60
3	Umidade de infiltração	Dano estético, abertura para novas anomalias.	Paredes expostas a intempéries	6	5	6	180
4	Fissuras, trincas, rachaduras, fendas e brechas	Oportunidade para umidade, incômodo.	Ausência de verga e contra-verga em aberturas, excesso de carga na cobertura.	3	2	5	30
5	Destacamento de pintura	Dano estético, incômodo.	Infiltração e uso	3	2	3	18
6	Destacamento de reboco	Dano estético, abertura para novas anomalias, incômodo.	Capilaridade	6	4	4	96
7	Biodeterioração	Dano estético, incômodo.	Parede exposta a intempéries.	4	4	4	64
8	Recalque diferencial	Dano estrutural	Afundamento do solo.	7	10	3	210

Fonte: Adaptado Regos (2022).

Tabela 11 - Resultado obtido pela hierarquização em ordem decrescente

Manifestação patológica	NPR	Prioridade
Recalque diferencial	210	1
Umidade de infiltração	180	2
Destacamento de reboco	96	3
Biodeterioração	64	4
Umidade ascensional	60	5
Umidade accidental	36	6
Fissura, trinca, rachadura, fenda, brecha	30	7
Destacamento de pintura	18	8

Fonte: Adaptado Regos (2022).

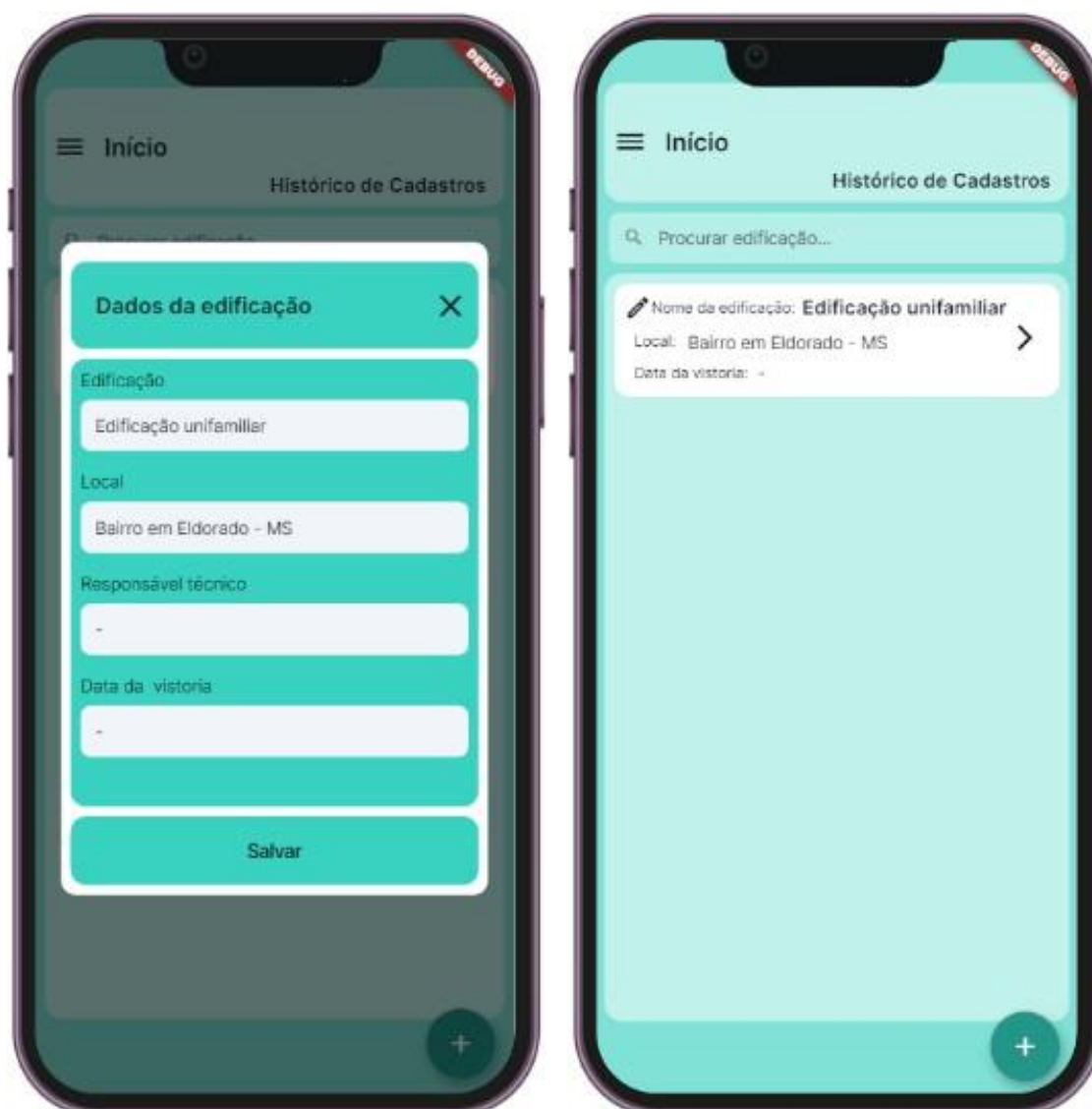
Para o teste de funcionalidade do protótipo, foram utilizados os dados coletados no trabalho da autora, os quais foram inseridos no aplicativo, obtendo-se os seguintes resultados, ilustrados nas figuras. Primeiramente, foi criada uma conta para o acesso ao aplicativo, conforme a Figura 71.

Figura 71 - Criação da conta no aplicativo

Fonte: Elaborado pelo autor.

Com acesso finalizado, na tela de Início, cadastrou-se a edificação em estudo de acordo com as informações fornecidas por Regos (2022).

Figura 72 - Sequência de telas para o cadastro da edificação



Fonte: Elaborado pelo autor.

Posteriormente, acessou-se a tela referente ao cadastro das manifestações patológicas. Nela, foram inseridas as mesmas informações da tabela do trabalho de Regos (2022), obtendo os resultados já hierarquizados na interface gráfica do aplicativo, com os valores de NPR em ordem decrescente, conforme o apresentado nas figuras a seguir.

Figura 73 - Cadastro das manifestações patológicas

The image shows a smartphone screen with the 'Matriz FMEA' app. The app has a dark green header with a back arrow and the title 'Matriz FMEA'. Below the header, the 'Nome da Edificação' is set to 'Edificação unifamiliar'. The main section is titled 'Manifestação' and contains a form for recording pathological manifestations. The form fields are as follows:

- Dados da manifestação patológica** (Section Header)
- Manifestação patológica**: Umidade acidental
- Local**: Parede externa da edificação
- Efeito**: Dano estético, abertura para novas anomalias
- Causa**: Águas pluviais, falta de rufos
- Severidade (S)**: 4
- Ocorrência (O)**: 3
- Deteção (D)**: 3
- Ação corretiva**: Não definida
- Foto**: (Icon for adding a photo)

At the bottom of the form are two buttons: 'Fechar' (Close) and 'Salvar' (Save). A green circular button with a plus sign is located at the bottom right of the screen.

Fonte: Elaborado pelo autor

Figura 74 - Visualização geral da tabela no aplicativo

← Matriz FMEA

Nome da Edificação: Edificação unifamiliar

Manifestação patológica	Local/Foto	Efeito	Causa	S	O	D	NPR	Ação corretiva	Check /Editar
Recalque diferencial	Paredes internas e externas da edificação	Dano estrutural	Afundamento do solo	7	10	3	210	Ação não definida	☐
Umidade de infiltração	Paredes internas da edificação	Dano estético, abertura para novas anomalias	Paredes expostas à intempéries	6	5	6	180	Ação não definida	☐
Destacamento do reboco	Paredes externas da edificação	Dano estético, abertura para novas anomalias, incômodo	Capilaridade	6	4	4	96	Ação não definida	☐
Biodeterioração	Paredes externas da edificação	Dano estético, incômodo	Parede exposta à intempéries	4	4	4	64	Ação não definida	☐
Umidade ascensional	Paredes externas da edificação	Dano estético, abertura para novas anomalias	Falta de impermeabilização das vigas baldrame	5	3	4	60	Ação não definida	☐

Rows per page: 10 1-6 of 6

Gerar pdf

+

← Matriz FMEA

Nome da Edificação: Edificação unifamiliar

Manifestação patológica	Local/Foto	Efeito	Causa	S	O	D	NPR	Ação corretiva	Check /Editar
Biodeterioração	edificação	Dano estético, incômodo	Parede exposta à intempéries	4	4	4	64	Ação não definida	☐
Umidade ascensional	Paredes externas da edificação	Dano estético, abertura para novas anomalias	Falta de impermeabilização das vigas baldrame	5	3	4	60	Ação não definida	☐
Umidade acidental	Parede externa da edificação	Dano estético, abertura para novas anomalias	Águas pluviais, falta de rufos	4	3	3	36	Ação não definida	☐
Fissura, trinca, rachadura, fenda, brecha	Paredes internas e externas da edificação, próximas à aberturas de portas e janelas	Oportunidade para umidade, incômodo	Ausência de verga e contra verga em aberturas, excesso de carga na cobertura	3	2	5	30	Ação não definida	☐
Destacamento da pintura	Paredes internas e externas da edificação	Dano estético, incômodo	Infiltração e uso	3	2	3	18	Ação não definida	☐

Rows per page: 10 1-6 of 6

Gerar pdf

+

Fonte: Elaborado pelo autor.

Figura 75 – Visualização das informações na tela do celular

Nome da Edificação: Edificação unifamiliar

Manifestação patológica	Local/Foto	
Recalque diferencial	Paredes internas e externas da edificação	Da
Umidade de infiltração	Paredes internas da edificação	De abert
Destacamento do reboco	Paredes externas da edificação	De abert anom
Biodeterioração	Paredes externas da edificação	De
Umidade	Paredes externas da edificação	Da

Rows per page: 10 1-8 of 8

Gerar pdf

+

Nome da Edificação: Edificação unifamiliar

	D	NPR	Ação corretiva	Check /Editar
	3	210	Ação não definida	☐
	6	180	Ação não definida	☐
	4	96	Ação não definida	☐
	4	64	Ação não definida	☐
				☐

Rows per page: 10 1-8 of 8

Gerar pdf

+

Fonte: Elaborado pelo autor.

Com todas as informações inseridas no aplicativo, já é possível visualizar na interface a hierarquização das manifestações patológicas em ordem decrescente. Após essa análise, comparou-se com a tabela final de Regos (2022), confirmando a consistência e a precisão dos resultados encontrados. Algumas informações não foram inseridas, como a ação corretiva, já que não é o foco do trabalho da autora. Em seguida, um documento PDF foi gerado, reunindo todos os dados fornecidos pelo usuário, conforme as Figura 76 e 77. As informações exibidas no banco de dados estão detalhadas na Figuras 78.

Figura 76 - Documento PDF gerado

Hierarquização das manifestações patológicas

Nome da Edificação: Edificação unifamiliar
 Local: Bairro em Eldorado - MS
 Responsável pela vistoria: -
 Data da vistoria: Não informada

Matriz FMEA - Página 1 de 4

Manifestação Patológica	Local/Foto	Efeito	Causa	S	O	D	NPR	Ação corretiva
1° Recalque diferencial	Paredes internas e externas da edificação <i>Sem foto</i>	Dano estrutural	Afundamento do solo	7	10	3	210	Ação não definida
2° Umidade de infiltração	Paredes internas da edificação <i>Sem foto</i>	Dano estético, abertura para novas anomalias	Paredes expostas à intempéries	6	5	6	180	Ação não definida

Hierarquização das manifestações patológicas

Nome da Edificação: Edificação unifamiliar
 Local: Bairro em Eldorado - MS
 Responsável pela vistoria: -
 Data da vistoria: Não informada

Matriz FMEA - Página 2 de 4

Manifestação Patológica	Local/Foto	Efeito	Causa	S	O	D	NPR	Ação corretiva
3° Destacamento do reboco	Paredes externas da edificação <i>Sem foto</i>	Dano estético, abertura para novas anomalias, incômodo	Capilaridade	6	4	4	96	Ação não definida
4° Biodegradação	Paredes externas da edificação <i>Sem foto</i>	Dano estético, incômodo	Parede exposta à intempéries	4	4	4	64	Ação não definida

Fonte: Elaborado pelo autor.

Figura 77 - Continuação do documento PDF gerado

Hierarquização das manifestações patológicas

Nome da Edificação: Edificação unifamiliar
 Local: Bairro em Eldorado - MS
 Responsável pela vistoria: -
 Data da vistoria: Não informada

Matriz FMEA - Página 3 de 4

Manifestação Patológica	Local/Foto	Efeito	Causa	S	O	D	NPR	Ação corretiva
5° Umidade ascensional	Paredes externas da edificação <i>Sem foto</i>	Dano estético, abertura para novas anomalias	Falta de impermeabilização das vigas baldramas	5	3	4	60	Ação não definida
6° Umidade acidental	Parede externa da edificação <i>Sem foto</i>	Dano estético, abertura para novas anomalias	Águas pluviais, falta de rufos	4	3	3	36	Ação não definida

Hierarquização das manifestações patológicas

Nome da Edificação: Edificação unifamiliar
 Local: Bairro em Eldorado - MS
 Responsável pela vistoria: -
 Data da vistoria: Não informada

Matriz FMEA - Página 4 de 4

Manifestação Patológica	Local/Foto	Efeito	Causa	S	O	D	NPR	Ação corretiva
7° Fissura, trinca, rachadura, fenda, brecha	Paredes internas e externas da edificação, próximas à aberturas de portas e janelas <i>Sem foto</i>	Oportunidade para umidade, incômodo	Ausência de verga e contra verga em aberturas, excesso de carga na cobertura	3	2	5	30	Ação não definida
8° Destacamento da pintura	Paredes internas e externas da edificação <i>Sem foto</i>	Dano estético, incômodo	Infiltração e uso	3	2	3	18	Ação não definida

Fonte: Elaborado pelo autor.

Figura 78 - Informações exibidas no banco de dados

schema: public

+ New table

Search tables...

Dadosdaedificacao

Dadosmanifestacaopatolo...

Usuario

Filter

Sorted by 1 rule

Insert

Auth policy

role postgres

Realtime on

API Docs

	id int8	id_dadosdaedificacao int8	nomedapatologia text	localdapatologia text	efeito text	causa text
	277	37	Recalque diferencial	Paredes internas e externas da edificação	Dano estrutural	Afundamento do solo
	267	37	Umidade de infiltração	Paredes internas da edificação	Dano estético, abertura para novas anomalias	Paredes expostas à intempéries
	268	37	Destacamento do reboco	Paredes externas da edificação	Dano estético, abertura para novas anomalias, inc	Capilaridade
	270	37	Biodeterioração	Paredes externas da edificação	Dano estético, incômodo	Parede exposta à intempéries
	266	37	Umidade ascensional	Paredes externas da edificação	Dano estético, abertura para novas anomalias	Falta de impermeabilização das vigas baldrames
	265	37	Umidade acidental	Parede externa da edificação	Dano estético, abertura para novas anomalias	Águas pluviais, falta de rufos
	272	37	Fissura, trinca, rachadura, fenda, brecha	Paredes internas e externas da edificação	Oportunidade para umidade, incômodo	Ausência de verga e contra verga em aberturas, excesso de carga
	269	37	Destacamento da pintura	Paredes internas e externas da edificação	Dano estético, incômodo	Infiltração e uso

schema: public

+ New table

Search tables...

Dadosdaedificacao

Dadosmanifestacaopatolo...

Usuario

Filter

Sorted by 1 rule

Insert

Auth policy

role postgres

Realtime on

API Docs

	id int8	s int4	o int4	d int4	npr int4	acaocorretiva text	check bool	status text	
	277	solo	7	10	3	210	Ação não definida	NULL	Em andamento
	267	à intempéries	6	5	6	180	Ação não definida	NULL	Em andamento
	268		6	4	4	96	Ação não definida	NULL	Em andamento
	270	intempéries	4	4	4	64	Ação não definida	NULL	Em andamento
	266	abilização das vigas baldrames	5	3	4	60	Ação não definida	NULL	Em andamento
	265	lta de rufos	4	3	3	36	Ação não definida	NULL	Em andamento
	272	a e contra verga em aberturas, excesso de carga na cobertura	3	2	5	30	Ação não definida	NULL	Em andamento
	269		3	2	3	18	Ação não definida	NULL	Em andamento

Fonte: Elaborado pelo autor.

5 DISCUSSÃO

A metodologia FMEA, dentre as demais abordagens para o gerenciamento de riscos nas atividades de Inspeção Predial, permite uma análise aprofundada tanto das falhas recorrentes, determinando os riscos mais críticos, quanto das condições que podem desencadear a ocorrência de falhas futuras, baseando-se nos padrões de manifestações observados. No entanto, o processo de hierarquização e organização dos dados coletados, quando aplicado em sistemas mais complexos, torna o gerenciamento das informações mais lento e difícil em processos realizados manualmente, podendo afetar a consistência das informações. Assim, o aplicativo móvel possibilita que essa gestão e obtenção dos dados, como os valores de NPR, seja mais eficaz e precisa. Somado a isso, a solução reúne históricos que podem ser consultados tanto no dispositivo móvel quanto no banco de dados, o que facilita a interação entre o ambiente de trabalho no campo e escritório.

Com o objetivo de garantir a compatibilidade e a liberação para todos os sistemas operacionais, toda a arquitetura do aplicativo foi projetada, e as tecnologias foram escolhidas para garantir que a solução possa receber atualizações e adaptações, tornando-a mais robusta conforme as necessidades. A plataforma *FlutterFlow* simplifica em muitos aspectos o desenvolvimento de aplicativos, ao trazer a perspectiva do *low-code* como ponto estratégico. A escolha dessa tecnologia para a criação de uma solução funcional, com integração a um banco de dados e facilidade de escalabilidade, tornou o processo otimizado e sem a necessidade de codificar cada componente de tela. Isso possibilitou a construção de telas com *layouts* responsivos e intuitivos, de fácil navegação e acesso às funcionalidades da ferramenta. Além disso, viabilizou a criação de um aplicativo multiplataforma, ou seja, projetado para funcionar em vários sistemas operacionais, o que torna seu uso mais amplo e de maior alcance.

No entanto, devido à plataforma de desenvolvimento ser uma tecnologia recente, ela apresenta alguns desafios, especialmente para usuários com pouca familiaridade com a programação de *softwares*. Entre os desafios encontrados, destaca-se a limitação de recursos com suporte em português, já que os fóruns e tutoriais em vídeo estão disponíveis, predominantemente, em outros idiomas. Ainda assim, houve dificuldades em encontrar respostas específicas para dúvidas mais complexas. Também foram identificados desafios na criação do fluxo de informações

entre as telas, assim como na implementação de lógicas integradas com o banco de dados e códigos customizados, o que exige um nível de conhecimento técnico mais avançado.

Por meio da aplicação da ferramenta virtual no estudo de caso, foi possível perceber a agilidade no processamento dos dados. O tempo gasto para hierarquizar e calcular os riscos foi menor do que a realização desse processo manualmente, que demandaria um tempo considerável devido à necessidade de alimentação e organização de planilhas. Outra vantagem é que as manifestações patológicas podem ser tratadas de forma individualizada, sem a necessidade de agrupá-las por repetição, o que poderia ter sido feito por Regos (2022), caso o aplicativo já estivesse disponível. Além disso, todas as tabelas elaboradas e apresentadas em seu trabalho seriam sintetizadas em único documento, com fotos anexadas diretamente no formulário FMEA. Isso permite uma análise mais detalhada e eficiente de cada manifestação patológica, garantindo um foco mais preciso no diagnóstico e na definição das ações corretivas necessárias.

Dessa forma, o aplicativo reduz o fluxo de trabalho, já que realiza automaticamente esses cálculos e gerencia as informações, apresentando os resultados de forma organizada em tempo real e com a possibilidade de serem sintetizados em documento PDF. É importante ressaltar que o aplicativo serve como uma ferramenta auxiliar no processo decisório. Sendo assim, não exige a necessidade de conhecimento técnico, e sua utilização deve ser feita por profissionais qualificados e as informações obtidas devem ser complementadas com normas e diretrizes.

O trabalho desenvolvido não está isento de limitações. O aplicativo requer a conexão com a internet no dispositivo móvel para que os dados sejam salvos corretamente no banco de dados. Essa limitação pode dificultar o uso em áreas onde não há acesso à rede *Wi-Fi* ou com sinal de internet fraco. Em contrapartida, melhorias nesse quesito podem ser feitas com o desenvolvimento de versões futuras que permitem que aplicativo funcione no modo *offline*.

6 CONSIDERAÇÕES FINAIS

O aplicativo cumpre o objetivo de otimizar a aplicação da metodologia FMEA. O desenvolvimento do aplicativo *low-code* configura-se como abordagem viável para auxiliar o processo de hierarquização das manifestações patológicas identificadas durante a inspeção predial. A metodologia FMEA, aplicada nesse contexto e integrada a uma interface gráfica, torna seu uso mais ágil e intuitivo, ao proporcionar recursos visuais e interações simplificadas entre as telas do aplicativo, além de funcionalidades práticas como gerar documentos em PDF com as informações inseridas.

Essa ferramenta serve como suporte aos profissionais em atividades de inspeções, promovendo a organização automática dos dados coletados. O aplicativo demonstrou ser uma solução eficaz, agregando a praticidade no tratamento das informações obtidas e possibilitando uma tomada de decisões mais fundamentada em dados consistentes. Além disso, facilita a realização de consultas futuras, o que permite *insights* valiosos e o acompanhamento contínuo do edifício e das ações corretivas implementadas. Dessa forma, a elaboração do planejamento de manutenções prediais torna-se mais eficiente, e quando aliado às inspeções periódicas, o aplicativo desempenha um papel preventivo, fundamental para reduzir custos com ações corretivas não planejadas.

Os resultados alcançados por meio da aplicação no estudo de caso, foram promissores e abrem caminhos para novas abordagens e funcionalidades que poderão ser incorporadas à ferramenta. Apesar de ser voltado para inspeções prediais em edificações, a estrutura do aplicativo pode ser adaptada para outros contextos dentro da Engenharia civil, como gestão riscos ambientais e de infraestrutura. Dessa forma, o desenvolvimento do protótipo do aplicativo MAniFE incentiva a digitalização de processos e mostra as possibilidades para a criação de soluções técnicas e práticas, estimulando a comunidade acadêmica e promovendo a colaboração interdisciplinar entre as áreas da engenharia e da tecnologia.

7 RECOMENDAÇÕES

Como o aplicativo está em sua fase alfa de desenvolvimento, ou seja, fase de teste em um ambiente controlado que envolve a correção de problemas de funcionalidades, recomenda-se que sejam realizados testes com um grupo seletivo de usuários externos para a coleta de *feedbacks*. O aplicativo pode ser ainda aprimorado com uma versão corporativa, voltada para organizações, através do controle de acesso por e-mails corporativos válidos, garantindo maior segurança das informações. Quanto às funcionalidades do aplicativo, pode-se implementar relações mais robustas com banco de dados, tratando outros parâmetros, como análises financeiras dos custos das intervenções.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALVES, G. P. **Sistemas de pintura em edifícios públicos de Maringá: Patologias, processos, execução e recomendações**. 2010. 60 f. Monografia (Especialização em Construção de Obras Públicas no Curso) – Universidade Federal do Paraná, Maringá, 2010.
- ANTUNES, G. R. **Estudo de manifestações patológicas em revestimento de fachada em Brasília: sistematização da incidência de casos**. 2010. 199 f. Dissertação (Mestrado em Estruturas e Construção Civil) – Universidade de Brasília, Brasília, 2010.
- ARAÚJO, C. S. **Análise de falhas, na fase de projeto, que podem comprometer a qualidade na construção civil: um estudo de caso, em uma empresa júnior, utilizando FMEA**. 2019. 61 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Civil) – Universidade Tecnológica Federal do Paraná, Campo Mourão, 2019.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 6118**: Projeto de estruturas de concreto – Procedimento. Rio de Janeiro: ABNT, 2023.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 15575**: Desempenho de edificações habitacionais. Rio de Janeiro: ABNT, 2024.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 16747**: Inspeção Predial – Diretrizes, conceitos, terminologia e procedimento. Rio de Janeiro: ABNT, 2020.
- BARSALOU, Matthew. **Investigation into a potential reduction of fmea efforts using action priority**. Management and Production Engineering Review, v. 13, n. 4, 2022.
- BIGOLIN, M.; PACHECO, L. S.; SILVA FILHO, L. C. P. **Inspeção predial e norma de desempenho: agentes intervenientes**. In: Anais do 56º Congresso Brasileiro do Concreto, CBC 2014, Rio Grande do Sul, 2014.
- BRITO, Thaís Farias de. **Análise de manifestações patológicas na construção civil pelo método GUT: Estudo de caso em uma instituição pública de ensino superior**. 2017. 77 f. Monografia (Especialização) – Universidade Federal da Paraíba, João Pessoa, 2017.
- BITTENCOURT, Náfez Souza. **Análises de manifestações patológicas na construção civil: um estudo de caso em uma instituição pública de pesquisa**. 2021. 79 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Civil) – Faculdade Maria Milza (FAMAM), Governador Mangabeira, 2021.
- CRONAPP. **Low-code e aplicativos mobile: entenda um pouco mais sobre!**. Blog Cronapp, 2023. Disponível em: <<https://blog.cronapp.io/low-code-mobile/>>. Acesso em: 09 dez. 2024.

FIGUEIREDO, C. R.; CAVALCANTI, M. S. C.; QUINTANILHA, D. S. A. **Gestão da qualidade segundo registros de assistência técnica em edifícios**. Caderno Pedagógico, [S. l.], v. 21, n. 3, p. e3137, 2024. DOI: 10.54033/cadpedv21n3-060. Disponível em: <https://ojs.studiespublicacoes.com.br/ojs/index.php/cadped/article/view/3137>. Acesso em: 21 nov. 2024.

FLUTTERFLOW. **FlutterFlow: build beautiful apps visually**. FlutterFlow, 2024. Disponível em: <https://www.flutterflow.io/>. Acesso em: 15 dez. 2024.

GARTNER. **Gartner says cloud will be the centerpiece of new digital experiences**. Gartner Newsroom, Stamford, 2021. Disponível em: <https://www.gartner.com/en/newsroom/press-releases/2021-11-10-gartner-says-cloud-will-be-the-centerpiece-of-new-digital-experiences>. Acesso em: 08 dez. 2024.

GOMIDE, Tito L. F., PUJADAS, Flávia Z. A., NETO, Jerônimo C. P. **Técnicas de inspeção e manutenção predial: vistorias técnicas, check-up predial, normas comentadas, manutenção x valorização patrimonial e análise de risco**. 1. ed. São Paulo: Pini, 2006. ISBN 8572661727, 9788572661720.

GOMIDE, Tito L. F.; NETO, Jerônimo C. P. F; GULLO, Marco Antonio. **Engenharia diagnóstica em edificações**. São Paulo: Pini, 2009. 418 p. ISBN 9788572661928.

HAWKINS, P. G.; WOOLLONS, D. J. **Failure modes and effects analysis of complex engineering systems using functional models**. Artificial Intelligence in Engineering, v. 12, n. 4, p. 375-397, 1998. ISSN 0954-1810.

IBM. **Tipos de aplicativos móveis**. IBM Documentation, 2024. Disponível em: <https://www.ibm.com/docs/pt-br/maas360?topic=apps-mobile-app-types>. Acesso em: 08 dez. 2024.

INSTITUTO BRASILEIRO DE AVALIAÇÕES E PERÍCIAS DE ENGENHARIA. **Norma de Inspeção Predial Nacional**. São Paulo, 2012.

INSTITUTO BRASILEIRO DE AVALIAÇÕES E PERÍCIAS DE ENGENHARIA DE SÃO PAULO. **Inspeção Predial: a saúde dos Edifícios**. 2.ed. São Paulo, 2015.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO). **General principles on the design of structures for durability**. ISO 13823. 2008.

KARDEC, A.; NASCIF, J. **Manutenção: função estratégica**. 4. ed. Rio de Janeiro: Qualitymark, 2012.

LEAL, Fabiano; DE PINHO, Alexandre Ferreira; DE ALMEIDA, Dagoberto Alves. **Análise de falhas através da aplicação do FMEA e da Teoria Grey**. Revista Gestão Industrial, v. 2, n. 1, 2006.

LEITE, José Henrique de Carvalho. **Análise de manifestações patológicas em edifícios históricos no município de Governador Dix-Sept Rosado-RN**. 2019. 69 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Civil) –

Universidade Federal Rural do Semi-árido, Caraúbas, 2019.

MACHADO, Kethllyen Miranda. **Levantamento de Patologias Causadas por Umidade nas Edificações da Cidade de Manaus**. 2019. 20 f. Centro Universitário do Norte, Manaus, 2019.

MAKADIA, Harsh. **Low-code/no-code development: everything you need to know**. Maruti Techlabs, 2024. Disponível em: <https://marutitech.com/low-code-no-code-development/>. Acesso em: 09 dez. 2024.

MATOS, R. B.; MILAN, M. **Aplicação sistêmica do modo de análise de falhas e efeitos (FMEA) para o desenvolvimento de indicadores de desempenho de empresas de pequeno porte**. Revista Árvore, v. 33, p. 977-985, 2009.

MCCAIN, Cecelia. **Using an FMEA in a service setting**. Quality Progress, v. 39, n. 9, p. 24, 2006.

MOURA, Cândido. **Análise de modo e efeitos de falha potencial. (FMEA)**. Manual de Referência, p. 1-44, 2000.

MZOUGUI, Ilyas; EL FELSOUI, Zoubir. **Proposition of a modified FMEA to improve reliability of product**. Procedia Cirp, v. 84, p. 1003-1009, 2019.

OLIVEIRA, Alexandre Magno. **Fissuras e rachaduras causadas por recalque diferencial de fundações**. 2012. 96f. Monografia (Especialização em Gestão em Avaliações e Perícias) – Universidade Federal de Minas Gerais, Minas Gerais, 2012.

OLIVEIRA, Daniel Ferreira. **Levantamento de causas de patologias na construção civil**. 2013. 107 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Civil) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2013.

ORTIZ, José Francisco Barbosa. **A importância de inspeção predial periódica**. Conselho Regional de Engenharia e Agronomia de Mato Grosso, 2015. Disponível em: <https://www.crea-mt.org.br/porta1/a-importancia-de-inspecao-predial-periodica-5/>. Acesso em: 25 out. 2024.

PERES, Rosilena Martins. **Levantamento e Identificação de Manifestações Patológicas em Prédio Histórico – Um Estudo de Caso**. 2001. 158 f. Dissertação (Mestrado) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2001.

PROJECT MANAGEMENT INSTITUTE. **Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos (PMBOK Guide)**. 5. ed. Newtown Square: Project Management Institute, 2013.

REGOS, E. C. **Análise de manifestações patológicas em alvenaria de vedação vertical de uma edificação residencial unifamiliar com a utilização da metodologia Failure Mode and Effect Analysis – FMEA**. 2022. 38 f. Universidade Paraense, Guaíra, 2022.

SAKURADA, E. Y. **As técnicas de Análise do Modos de Falhas e seus Efeitos e**

Análise da Árvore de Falhas no desenvolvimento e na avaliação de produtos. 2001. 145 f. Dissertação (Mestrado) – Universidade Federal de Santa Catarina, Florianópolis, 2001.

SANCHES, C.E. **Failure mode and effect analyze (FMEA).** 2010. Disponível em: <https://pt.scribd.com/doc/80226992/Apostila-Fmea-e-Analise-Do-Valor-2010>. Acesso em: 31 out. de 2024.

SANTOS, Matheus Medeiros dos. **Análise quantitativa das patologias em prédios da UFERSA utilizando a metodologia GDE/UNB.** 2017. 96 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Civil) – Universidade Federal Rural do Semi-árido, Mossoró, 2017.

SANTOS, C. R. B.; SILVA, D. L.; NASCIMENTO, I. M. S. **Incidência de Manifestações Patológicas em Edificações Residenciais na Região Metropolitana do Recife (RMR).** Revista de Engenharia e Pesquisa Aplicada, [S.l.], v. 2, n. 3, 2017. p. 76-83.

SCHNEIDER, Aline Lessa da Silva. **A importância do gerenciamento de risco em projetos.** 2014. 15 f. Centro de Ensino Unificado de Brasília, Brasília, 2014.

SILVA, Arthur Bastos. **Aplicação de método combinado de ferramentas de gestão da qualidade FMEA e FTA na detecção e controle de patologias prediais.** 2015. 134 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Civil) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2015.

SILVA, Débora Luana Henrique Soares. **Aplicação do FMEA como suporte para melhoria de processos na construção civil: um estudo de caso.** 2019. 52 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Produção) – Universidade Federal de Campina Grande, Sumé, 2019.

SILVA, Fransérgio Aparecido de Souza; PRADO, Ely Fernando do. **Análise teórica sobre o desenvolvimento de aplicativos nativos, híbridos e webapps.** 2019. Revista EduFatec: educação, tecnologia e gestão, Franca, p. 1-18, v.2 n.1. Disponível em: <https://revistaedufatec.fatecfranca.edu.br/wp-content/uploads/2019/09/AN%C3%81LISE-TE%C3%93RICA-SOBRE-O-DESENVOLVIMENTO-DE-APLICATIVOS-NATIVOS-H%C3%84BRIDOS-E-WEBAPPS.pdf>. Acesso em: 08 dez. 2024.

SILVA, José Lucas Belarmino da. **Uma análise do desenvolvimento das linguagens low code e suas perspectivas de introdução e contribuição no âmbito educacional.** 2023. 48 f. Trabalho de Conclusão de Curso (Licenciatura em Computação) – Universidade Federal de Paraíba, João Pessoa, 2023.

SOMMERVILLE, Ian. **Engenharia de software.** Tradução Ivan Bosnic e Kalinka G. de O. Gonçalves; revisão técnica Kechi Hirma. 9. ed. São Paulo: Pearson Prentice Hall, 2011. Título original: Software engineering. ISBN 978-85-7936-108-1.

STAMATIS, D. H. **Failure Mode Effect Analysis: FMEA from Theory to Execution.** Milwaukee: American Society for Quality, Quality Press. 2003. Disponível em:

<https://books.google.com.br/books?id=TTxl8jbTkVwC>. Acesso em: 22 out. 2024.

SUPABASE. **Supabase: open source Firebase alternative**. Supabase, 2024. Disponível em: <https://supabase.com/>. Acesso em: 15 dez. 2024.

TAKAYAMA, M. A. S. **Análise de falhas aplicada ao planejamento estratégico da manutenção**. 2008. 57 f. Trabalho de conclusão de curso (Engenharia de Produção). Universidade Federal de Juiz de Fora, 2008.

THOMAZ, E. Trincas em edifícios causas, prevenção e recuperação. PINI:Escola Politécnica da Universidade de São Paulo: Instituto de Pesquisas Tecnológicas, 1ª ed. São Paulo, 1989.

TOLEDO, J. C.; AMARAL, D. C. **FMEA – Análise do tipo e efeito de falha**. GEPEQ /UFSCar, 2006. Disponível em: <https://www.smar.net.com.br/qualidade/FerramentasQualidade/FMEA-APOSTILA.pdf>. Acesso em: 01 set. 2024.

TUTIKIAN, B.; PACHECO, M. **Boletín Técnico 1:inspección, diagnóstico y pronóstico en la construcción civil**. Mérida, México: Asociación Latinoamericana de Control de Calidad, Patología y Recuperación de la Construcción, ALCONPAT, 2013.