UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA MECÂNICA

CURSO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA

RICARDO OLIVEIRA DOS SANTOS JÚNIOR

DESENVOLVIMENTO E APLICAÇÃO DE UM SISTEMA DE GERENCIAMENTO DE MANUTENÇÃO "*NO CODE*" EM BUBBLE

FORTALEZA

2023

RICARDO OLIVEIRA DOS SANTOS JÚNIOR

DESENVOLVIMENTO E APLICAÇÃO DE UM SISTEMA DE GERENCIAMENTO DE MANUTENÇÃO "*NO CODE*" EM BUBBLE

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Mecânica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia Mecânica.

Orientador: Prof. Dr. Francisco Elicivaldo Lima

FORTALEZA

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S238d Santos Júnior, Ricardo Oliveira dos.
Desenvolvimento e aplicação de um sistema de gerenciamento de manutenção “no code” em bubble / Ricardo Oliveira dos Santos Júnior. – 2023.
49 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia Mecânica, Fortaleza, 2023.
Orientação: Prof. Dr. Francisco Elicivaldo Lima.

1. Manutenção. 2. Gerenciamento. 3. Baixo código. 4. Desenvolvimento. I. Título.

CDD 620.1

RICARDO OLIVEIRA DOS SANTOS JÚNIOR

DESENVOLVIMENTO E APLICAÇÃO DE UM SISTEMA DE GERENCIAMENTO DE MANUTENÇÃO "*NO CODE*" EM BUBBLE

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Mecânica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia Mecânica.

Orientador: Prof. Dr. Francisco Elicivaldo Lima

Aprovado em:

BANCA EXAMINADORA

Prof. Dr. Francisco Elicivaldo Lima (Orientador)

Universidade Federal do Ceará (UFC)

Prof. Dr. Francisco Ilson Silva Junior

Universidade Federal do Ceará (UFC)

Prof. Dr. Luiz Soares Junior

Universidade Federal do Ceará (UFC)

AGRADECIMENTOS

À Deus por me proporcionar o dom da vida e aprendizado durante toda essa caminhada.

Aos meus pais Lano, Silvia e Joana por todo o apoio durante os momentos mais difíceis, proporcionando tudo do bom e do melhor.

Aos meus avós Lenir e Expedito, em especial a minha falecida avó Lêda, que sempre me incentivou para correr atrás dos meus sonhos, e o primeiro passo para isso foi a graduação.

À Universidade Federal do Ceará por proporcionar toda a estrutura e preparação para o mercado de trabalho e a todos os excelentes professos com que tiver aula.

Aos professores membros da banca avaliadora, Prof. Dr. Luiz Soares Junior e Prof. Dr. Francisco Ilson Silva Junior, em especial ao meu orientador Prof. Dr. Francisco Elicivaldo Lima para dispor do seu tempo para a orientação nesse presente trabalho.

A todos os meus amigos e pessoas que contribuíram indiretamente durante essa desafiadora jornada de graduação.

A todos os meus colegas de trabalho da Santana Textiles que me deram total apoio e incentivo para a conclusão da graduação.

“Os céus declaram a glória de Deus e o firmamento anuncia a obra das suas mãos.”
Salmos 19:1

RESUMO

É imprescindível a coleta e padronização de registros de manutenção dentro de uma empresa, pois serão através deles que serão efetuadas as tomadas de decisões por parte do time de gestão. Para um melhor aproveitamento desse processo, empresas de grande porte utilizam-se de sistemas integrados ou terceirizados para a execução desse processo de suma importância. Contudo, nem todas as empresas podem arcar com os custos de licença desses softwares mencionados e outras soluções devem vim à tona. Com o crescimento da tecnologia e da automação, ferramentas de desenvolvimento de baixo código, que requerem pouca utilização de códigos ou nenhuma, estão ficando cada vez mais populares, visto que sua praticidade e eficiência são incomparáveis com o modo de desenvolvimento tradicional. Portanto, neste trabalho será apresentado uma tecnologia de desenvolvimento "*no code*" denominada "Bubble", e será desenvolvido uma aplicação para a coleta, padronização e organização desses registros de manutenção para, posteriormente, elaboração de indicadores que auxiliaram no planejamento de manutenção da empresa. O sistema foi aplicado em uma indústria têxtil no setor de fiação, em um tipo de máquina específico para elemento de fiar, "*Open End*". Por fim, o sistema teve resultados positivos no que diz respeito à redução de custos e potencial na entrega de resultados de indicadores.

Palavras-chave: Manutenção. Gerenciamento. Baixo código. Desenvolvimento.

ABSTRACT

The collection and standardization of maintenance records within a company are essential, as they will be the basis for decision-making by the management team. To optimize this process, large companies often employ integrated or outsourced systems to carry out this crucial task. However, not all companies can afford the licensing costs associated with these mentioned software solutions, so alternative options must be considered. With the growth of technology and automation, low-code development tools that require minimal or no coding are becoming increasingly popular due to their practicality and efficiency compared to traditional development methods.Therefore, this work will introduce a “no-code” development technology called “Bubble”, and an application will be developed for the collection, standardization, and organization of these maintenance records, which will later be used to create indicators to assist in the company's maintenance planning. The system was implemented in a textile industry within the spinning sector, specifically for a type of machine called “Open End” used for spinning elements. Finally, the system yielded positive outcomes concerning cost reduction and potential for delivering indicator results.

Keywords: Maintenance. Management. Low code. Development.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

CMMS	*Computerized Maintenance Management System* (Sistema Automatizado de Gerência da Manutenção)
MTBF	*Mean Time Between Failures* (Tempo médio entre falha)
MTTR	*Mean Time To Repair* (Tempo médio para reparo)
ABNT	Associação Brasileira de Normas Técnicas
NBR	Norma Técnica Brasileira
CS	*Custom State* (Estado personalizado)

SUMÁRIO

1 INTRODUÇÃO

1.1 Contextualização

A gestão da manutenção é um dos principais fatores para o bom funcionamento de uma indústria. Para tomar decisões informadas, os gestores precisam ter acesso a informações precisas e atualizadas. Nesse sentido, a implementação de um Sistema Automatizado de Gerência da Manutenção, ou comumente chamado de CMMS (*Computerized Maintenance Management System*) é de fundamental importância para organizar e produzir indicadores relevantes.

Uma plataforma de gerenciamento de registros de manutenção oferece uma abordagem sistematizada para coletar, armazenar e analisar dados relacionados às atividades de manutenção, permitindo o registro de maneira padronizada, facilitando a busca e a modelagem desses dados.

Empresas de pequeno e médio porte muitas vezes enfrentam desafios na gestão dos dados de manutenção, pois não possuem sistemas adequados para essa finalidade. Na realidade atual, esses sistemas de gerenciamento de manutenção, com uma tecnologia de análise avançada, estão disponíveis somente para uma pequena parte das empresas, geralmente para as de grande porte devido a fatores como alto custo para adesão desses *softwares* (COSTA, 2019).

Um dos principais problemas que as empresas enfrentam é a complexidade de suas operações. Para a manutenção não é diferente, e a necessidade de uma resposta rápida e flexível para situações de tomadas de decisões e adaptações por parte da equipe de acordo com a problemática, vão sendo cada vez mais imprescindíveis. Essas tomadas de decisões muita das vezes são condicionadas pela análise de dados estatísticos armazenados em um sistema de gerenciamento de manutenção, cuja função é possibilitar a compreensão de desempenho dos equipamentos e a identificação de tendências e padrões (COSTA, 2019).

Com o advento da pandemia, houve um aumento significativo no uso de plataformas de desenvolvimento web e mobile. Essas plataformas, também conhecidas como LCDP (*low-code development platform*), plataforma de desenvolvimento de baixo código, são projetadas para permitir que usuários sem habilidades de programação desenvolvam sistemas de maneira mais fácil e acessível (SANCHIS; PERALES; FRAILE; POLE, 2019).

Nesse contexto, neste trabalho apresenta solução para abordar o desafio de gestão de manutenção, considerando que sistemas robustos podem ser financeiramente inviáveis para

algumas empresas devido aos altos custos envolvidos, será proposto um sistema desenvolvido com base em uma plataforma "*low code*".

Essa abordagem permite que empresas superem a limitação financeira, desenvolvendo um sistema de gerenciamento de manutenção de forma mais econômica, visto que a facilidade e usabilidade dessas plataformas de desenvolvimento fomentam o investimento dessas empresas para a criação de um sistema personalizado sem a necessidade de conhecimento profundo em programação por parte do indivíduo que estará desenvolvendo.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver uma plataforma CMMS para gestão de registros de manutenção para uma indústria de pequeno ou médio porte.

1.2.2 Objetivos Específicos

a) Mostrar a viabilidade do desenvolvimento de um CMMS com uma tecnologia "*no code*";
b) Apresentar as funcionalidades no sistema e indicadores para a manutenção;
c) Identificar os requisitos para a elaboração de um CMMS desenvolvido em uma plataforma de baixo código;

1.3 Justificativa

A utilização de um sistema para controle de registros de manutenção é de fundamental importância para posteriormente traçar um planejamento. Para Kardec e Nascif (2009), esse sistema será responsável por determinar os seguintes aspectos, conforme a análise dos registros:

a) Quais serviços serão realizados;
b) Quando os serviços serão executados;
c) Quais recursos serão necessários para a realização dos serviços;
d) Quanto tempo será destinado a cada serviço;
e) Qual o custo de cada serviço;

Conforme ressaltado por Viana (2002), a importância de um sistema de manutenção abrange desde o cadastro das informações relacionadas a intervenções realizadas pelos

mantenedores até a análise dos relatórios gerados. Essa abordagem permite um controle efetivo e organizado de todas as atividades de manutenção.

Para Filho (2008), um CMMS contém uma grande quantidade de dados relevantes. A utilização desses registros contribui diretamente para o planejamento da manutenção, tornando-o mais ágil, eficiente e preciso. Essa otimização resulta em uma redução de custos para a empresa visto que possibilita uma melhor distribuição de recursos financeiros, humanos e materiais.

Dessa forma, o uso de um CMMS adequado proporciona benefícios significativos para a empresa, auxiliando no controle e na organização das intervenções, melhorando a alocação de recursos. Além disso, a análise dos relatórios gerados pelo sistema, auxilia a tomada de decisões estratégicas por parte dos gestores e a melhoria contínua dos processos de manutenção na empresa.

Ademais, a utilização de um CMMS desenvolvido por uma plataforma "*no code*" apresenta-se como uma solução viável e eficaz, visto que o baixo custo para o desenvolvimento é o principal fator somado com a capacidade de customização conforme as necessidades da empresa.

1.4 Metodologia

Para Gerhardt e Silveira (2009), os tipos de pesquisa possuem divisões bem definidas, nas quais um projeto pode se enquadrar. A pesquisa pode ser classificada em diferentes categorias com base em seus objetivos, métodos e abordagens utilizadas na pesquisa. A Figura 1 apresenta essas categorias conforme o seu tipo de pesquisa.

Figura 1- Tipos de pesquisa conforme a sua classificação

Classificação	Tipos de pesquisa
Quanto à finalidade	Pesquisa básica ou fundamental
	Pesquisa aplicada ou tecnológica
Quanto à natureza	Pesquisa observacional
	Pesquisa experimental
Quanto à forma de abordagem	Pesquisa qualitativa
	Pesquisa quantitativa
Quanto aos objetivos	Pesquisa exploratória
	Pesquisa explicativa
Quanto aos procedimentos técnicos	Pesquisa bibliográfica
	Pesquisa documental
	Pesquisa de laboratório
	Pesquisa de campo
Quanto ao desenvolvimento no tempo	Pesquisa transversal
	Pesquisa longitudinal
	Pesquisa prespectiva

Fonte: (GERHARDT; SILVEIRA, 2009), adaptado pelo autor.

O projeto tem como um de seus objetivos mostrar a viabilidade do desenvolvimento de um CMMS com uma tecnologia "*no code*" com uma aplicação prática, voltada para a solução de uma problemática, que é a redução de custos em uma empresa. Com isso, quanto a finalidade da pesquisa, caracteriza-se como uma pesquisa aplicada (GERHARDT; SILVEIRA, 2009).

Quanto à natureza, o projeto tem como objetivo gerar conhecimento para aplicações práticas no âmbito empresarial, mais especificamente na área de manutenção no que diz respeito à desenvolvimento de sistema. Isso o caracteriza como uma pesquisa experimental, pois busca desenvolver e testar um sistema de gerenciamento de manutenção com o objetivo de obter resultados aplicáveis e relevantes para a prática empresarial (GERHARDT; SILVEIRA, 2009).

Para o projeto, foi aplicado uma metodologia de análise de informação indutiva, com o foco na compreensão dos fenômenos e processos que caracterizam uma boa gestão de manutenção. Essa análise resulta em uma série de indicadores de manutenção no qual representam o principal ponto do projeto. Dessa forma, o projeto pode ser compreendido como uma abordagem tanto qualitativa como quantitativa (GERHARDT; SILVEIRA, 2009).

Quanto aos objetivos, o projeto caracteriza-se como exploratório. Isso ocorre porque uma das finalidades é explorar a problemática existente, que é a falta de um sistema de gerenciamento de manutenção e fornecer uma base de estudo para o desenvolvimento de softwares de manutenção com base em plataformas de desenvolvimento "*low code*" (GERHARDT; SILVEIRA, 2009).

Em relação aos procedimentos adotados, foram realizadas pesquisas bibliográficas sobre os conceitos da engenharia de manutenção que serão abordados no projeto. Além disso, foram conduzidas pesquisas documentais para analisar a plataforma de desenvolvimento utilizada no projeto. Esses procedimentos visam embasar teoricamente o desenvolvimento do CMMS e garantir uma abordagem consistente.

2 REVISÃO BIBLIOGRÁFICA

O objetivo deste capítulo é apresentar a fundamentação teórica relacionada à manutenção, proporcionando o embasamento necessário para a compreensão deste trabalho. Serão abordados os principais conceitos, teorias e modelos que sustentam a prática da manutenção, fornecendo um panorama completo sobre o tema.

2.1 Evolução da manutenção

A prática da manutenção já era empregada desde a Idade Média, quando artesãos desenvolviam instrumentos e ferramentas que, com o tempo, apresentavam desgaste e necessitavam de reparos para continuar sendo utilizados, ao invés de serem descartados por conta de uma falha (FILHO, 2008).

Com a chegada da revolução industrial no final do século XVIII, máquinas industriais de grande porte foram introduzidas nas indústrias, revolucionando a produção em grande escala. Contudo, nessa época não haviam equipe de mantenedores especializados nessa indústria para um eventual reparo grave na máquina, fazendo com que os fabricantes fossem efetuar o reparo ou troca de peças para solucionar o problema (FILHO, 2008).

Essa abordagem refletia a visão de que a manutenção era uma responsabilidade exclusiva do fabricante, uma vez que eles possuíam o conhecimento técnico e as peças de reposição necessárias para realizar os reparos. Dessa forma, as indústrias dependiam dos fabricantes para resolver problemas e garantir a continuidade da produção.

Para Kardec e Nascif (2009), a manutenção pode ser dividida em quatro gerações.

A primeira geração pode ser entendida como um período antes da Segunda Guerra Mundial, em um contexto em não havia uma priorização da produtividade por parte das indústrias da época e com isso, a manutenção não era vista com um grau de importância relevante. A manutenção em si, era feita somente de forma corretiva, após a quebra de uma componente (KARDEC; NASCIF, 2009).

Kardec e Nascif (2009) acreditavam que, à medida que as indústrias evoluíram e enfrentaram desafios durante a Segunda Guerra Mundial, começaram a surgir mudanças significativas na abordagem da gestão da manutenção. A necessidade de aumentar a produtividade, garantir a disponibilidade, maior confiabilidade dos equipamentos e otimizar os recursos levou ao desenvolvimento de novas estratégias e métodos de manutenção. Ademais, percebeu-se a importância de evitar a falha e ter uma previsibilidade da ferramenta, evitando o processo de corretiva, no qual resultou o conceito de manutenção preventiva, caracterizando assim, a segunda geração do processo evolutivo.

A terceira geração foi marcada pela influência do sistema "*just-in-time"*, que buscava minimizar os desperdícios e otimizar o fluxo de produção. Diante disso, percebeu-se que a melhoria dos processos estava diretamente relacionada ao nível de disponibilidade dos equipamentos, e que um grande número de falhas provocava consequências na segurança e no meio ambiente na indústria. Nesse contexto, surgiu-se a necessidade de uma abordagem mais avançada para a gestão da manutenção, que permitisse uma análise mais detalhada dos indicadores de disponibilidade e confiabilidade dos equipamentos acarretando no aumento da produção. Foi então introduzido o conceito de manutenção preditiva (KARDEC; NASCIF, 2009).

Com a chegada da quarta geração a partir da metade da década de 90, a manutenção já estava consolidada como uma parte primordial dentro de uma indústria visto que com a redução das falhas, resultaria em um aumento de produção. Uma das principais características da quarta geração foi a ênfase nos aspectos quantitativos dos indicadores de desempenho de manutenção, com uma maior utilização de instrumentação e sistemas de monitoramento para obtenção de informações precisas sobre a eficiência dos equipamentos. Com isso, o foco estava em medir e acompanhar os indicadores de eficiência, em contraste com a abordagem anterior, que se concentrava nos indicadores de ineficiência, como as falhas corretivas não planejadas e as intervenções preventivas. Vale salientar também que, foi implementado uma metodologia de integração entre as áreas de engenharia, manutenção e operação com o objetivo de otimizar o processo de produção (KARDEC; NASCIF, 2009).

Outro aspecto importante da quarta geração foi o avanço no uso de tecnologias de informação e comunicação na gestão da manutenção. A digitalização dos processos e a adoção de CMMS tornaram-se mais comuns pelo fato de que essas ferramentas possibilitaram um melhor controle das atividades de manutenção, o registro e análise de dados, a programação de intervenções e a geração de relatórios gerenciais, contribuindo com a melhoria da eficiência, o

aumento da confiabilidade e a redução dos custos de manutenção no geral (KARDEC; NASCIF, 2009).

A Figura 2 apresenta os marcos para cada geração mencionada, trazendo uma abordagem simples e objetiva

Figura 2- Evolução da manutenção

EVOLUÇÃO DA MANUTENÇÃO				
	Primeira Geração	Segunda Geração	Terceira Geração	Quarta Geração
Ano	1940 1950	1960 1970	1980 1990	2000 2010
Aumento das expectativas em relação à Manutenção	•Conserto após a falha	•Disponibilidade crescente •Maior vida útil do equipamento	•Maior confiabilidade •Maior disponibilidade •Melhor relação custo-benefício •Preservação do meio ambiente	•Maior confiabilidade •Maior disponibilidade •Preservação do meio ambiente •Segurança •Influir nos resultados do negócio •Gerenciar os ativos
Visão quanto à falha do equipamento	•Todos os equipamentos se desgastam com a idade e, por isso, falham	•Todos os equipamentos se comportam de acordo com a curva da banheira	•Existência de 6 padrões de falhas (Nowlan & Heap e Moubray) Ver Capítulo 5	•Reduzir drasticamente falhas prematuras dos padrões A e F. (Nowlan & Heap e Moubray) Ver Capítulo 5
Mudança nas técnicas de Manutenção	• Habilidades voltadas para o reparo	• Planejamento manual da manutenção • Computadores grandes e lentos • Manutenção Preventiva (por tempo)	• Monitoramento da condição • Manutenção Preditiva • Análise de risco • Computadores pequenos e rápidos • Softwares potentes • Grupos de trabalho multidisciplinares • Projetos voltados para a confiabilidade • Contratação por mão de obra e serviços	• Aumento da Manutenção Preditiva e Monitoramento da Condição • Minimização nas Manutenções Preventiva e Corretiva não Planejada • Análise de Falhas • Técnicas de confiabilidade • Manutenibilidade • Engenharia de Manutenção • Projetos voltados para confiabilidade manutenibilidade e Custo do Ciclo de Vida. • Contratação por resultados

Fonte: (KARDEC; NASCIF, 2009)

2.2 Tipos de manutenção

2.2.1 Manutenção Corretiva

Para Xenos (1998), a manutenção corretiva é sempre realizada após uma falha na máquina que ocasione em sua parada. Kardec e Nascif (2009) aborda como manutenção corretiva o ato de atuar para corrigir uma falha no equipamento, ou quaisquer sinais de desempenho abaixo do esperado. O autor aborda também que, diferentes de outros autores, como Xenos (1998), a manutenção corretiva não é necessariamente entendida como uma manutenção emergencial.

A manutenção corretiva pode ser classificada em duas categorias: planejada e não planejada. Na manutenção corretiva planejada, os gestores tomam a decisão de intervir em um equipamento específico com base nos sinais de baixa eficiência observados. Por outro lado, a manutenção corretiva não planejada ocorre de forma inesperada, resultando em uma parada não programada. Geralmente, a manutenção do tipo não planejada acarreta custos mais elevados, visto que na maioria dos casos há uma quebra por operação ou fadiga do equipamento (KARDEC; NASCIF, 2009).

A manutenção planejada apresenta uma série de vantagens em relação à não planejada, dentre elas podem considerar o menor custo de mão de obra e material, melhor condicionamento da máquina, otimização do fluxo de ações para efetuar a manutenção e a garantia de disponibilidade de peças de reposição, equipamentos e ferramentas necessárias (KARDEC; NASCIF, 2009).

Xenos (1998) faz uma comparação do ponto de vista de custos de manutenção, acreditando que a manutenção corretiva pode ser mais econômica do que realizar prevenções para não ocasionar a parada da máquina. Porém, vale salientar que os custos decorrentes da parada de produção também devem ser levados em consideração. Nesse sentido, a manutenção corretiva poderia sair com custos mais elevados do que uma não realização de prevenção de falhas no equipamento.

2.2.2 Manutenção Preventiva

Kardec e Nascif (2009) entende que a manutenção preventiva se caracteriza como o conjunto de ações que resultam na redução de falhas ou diminuição de quedas de desempenho da máquina, realizadas de forma antecipada. Essas ações são definidas em intervalos de tempo predeterminados em que são realizadas atividades de inspeção, lubrificação, ajustes e outras intervenções necessárias para evitar a degradação do equipamento.

Segundo Xenos (1998), a manutenção preventiva é o coração das atividades de manutenção e deve ser tratada de forma prioritária por qualquer empresa. Esse conjunto de ações acarretam em um aumento de eficiência dos equipamentos, além de contribuir para a redução de falhas e o aumento da disponibilidade e vida útil dos mesmos. Ademais, com a utilização de técnicas de manutenção preventiva, a empresa tem a possibilidade de um melhor controle dos tempos de parada dos equipamentos, impactando diretamente no planejamento da produção e na redução dos custos, minimizar interrupções na produção e aumentando a

confiabilidade dos equipamentos, resultando em uma maior eficiência operacional e ganhos financeiros a longo prazo.

Vale salientar que, pontos negativos com relação à manutenção preventiva devem ser apontados como a introdução de defeitos inexistentes devido a falha humana, falha por peça, contaminação de impurezas no óleo, falhas dos procedimentos de manutenção e danos no momento da partida ou parada da máquina (KARDEC; NASCIF, 2009).

2.2.3 Manutenção Preditiva

A manutenção preditiva foi uma quebra de paradigma no mundo da manutenção. A possibilidade de obter indicadores confiáveis que permitissem uma avaliação precisa do estado dos equipamentos, transformou a metodologia empregada pelas empresas nesse aspecto.

De acordo com Kardec e Nascif (2009), a manutenção preditiva envolve um conjunto de ações que são realizadas, levando em consideração parâmetros de condições ou desempenho que indicam as condições do equipamento, mesmo estando em funcionamento. Dessa forma, a possibilidade de ter uma visão mais precisa do estado de funcionamento do equipamento acarretou em um melhor planejamento para uma manutenção corretiva planejada visto que poderia ter uma ação de intervenção antecipada, melhorando a disponibilidade da máquina.

Xenos (1998) aponta que a manutenção preditiva costuma ser tratada como uma tecnologia muito avançada para esta nas mãos de qualquer pessoa dentro da empresa, e que essa forma de manutenção deve ser responsabilidade de pessoas como engenheiros ou indivíduos com um alto conhecimento técnico em manutenção.

2.2.4 Manutenção Detectiva

A manutenção detectiva consiste na atuação em sistemas de proteção com o objetivo de detectar falhas ocultas ou não perceptíveis por parte dos operadores ou da manutenção. Esse procedimento é realizado sem a necessidade de interromper o funcionamento dos equipamentos. Com isso, é permitido uma avaliação contínua e em tempo real, identificando possíveis falhas ou mau funcionamento dos sistemas de proteção (KARDEC; NASCIF, 2009).

2.3 Indicadores de manutenção

2.3.1 MTBF

O MTBF (*Mean Time Between Failures)*, está associado ao tempo médio entre falhas de um determinado equipamento relacionando o tempo total de operação do equipamento (TTO) pelo número de falhas (n).

$$MTBF = \frac{TTO}{n} \tag{1}$$

2.3.2 MTTR

O MTBF (*Mean Time To Repair*), consiste em um indicador que está diretamente ligado ao tempo médio para reparo de uma determinada ocorrência de manutenção. O cálculo relaciona o tempo total de manutenção corretiva (TTMC) pelo número de falhas (n).

$$MTTR = \frac{TTMC}{n} \tag{2}$$

2.3.3 Disponibilidade

A disponibilidade consiste no percentual em que a máquina ou determinada seção ficou em operação pelo tempo total que ela poderia ficar com um rendimento máximo incluindo o tempo total de manutenção preventiva (TTMP).

$$Disp = \frac{TTO}{TTO + TTMC + TTMP} \times 100\% \tag{3}$$

2.3.4 Confiabilidade

A confiabilidade consiste em indicador que traz a probabilidade de um equipamento ter uma falha em um determinado período de tempo (t). A variável (λ) corresponde a taxa de falhas em relação ao tempo.

$$Conf = e^{-\lambda t} \times 100\% \tag{4}$$

2.3.5 Diagrama de Pareto

O diagrama de Pareto um recurso gráfico que estabelece uma ordenação de falhas que devem ser sanados com base em um valor total acumulado. Esse indicador tem como objetivo priorizar as principais causas de falhas diante dos inúmeros outros.

2.4 Plataformas de baixo código

É notório que a falta de profissionais de programação ainda é um problema bastante recorrente no Brasil, visto que os profissionais dessa área optam por buscarem oportunidades em empresas do exterior. Com isso, entende-se a necessidade de empresas procurarem outras soluções de desenvolvimento para criarem ferramentas que ajudem em suas atividades diárias. Diante desse cenário, a automação da codificação surge como uma tendência, representando um avanço significativo na área da ciência da computação (SANCHIS; PERALES; FRAILE; POLE, 2019).

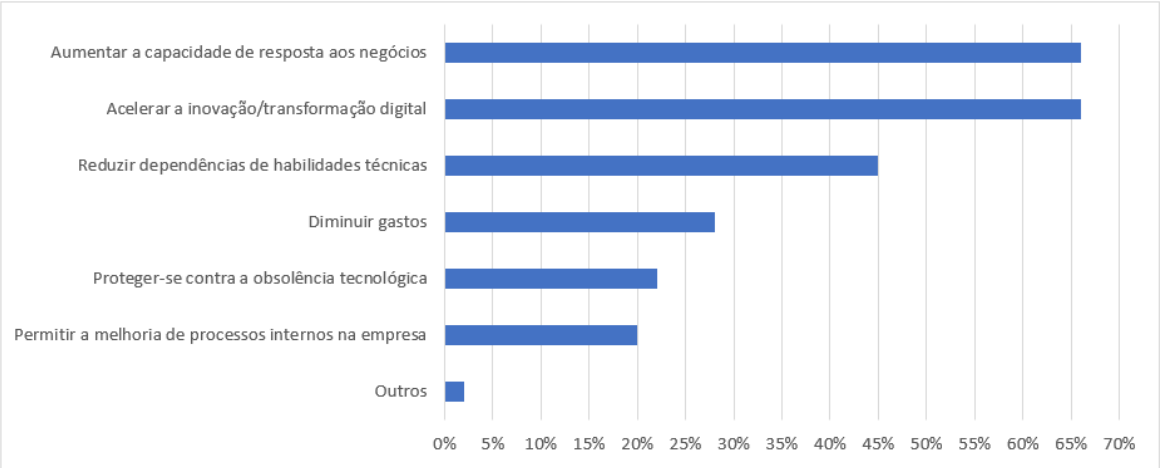
Essas ferramentas oferecem oportunidades para profissionais de diversas áreas desenvolverem soluções personalizadas conforme sua necessidade de maneira econômica e ágil sem comprometer a qualidade da operação.

O mercado de plataformas de desenvolvimento de baixo código tem prosperado nos últimos anos e várias empresas vem competindo nesse mercado para oferecerem uma melhor interface e funcionalidades para o usuário. Podemos citar alguns desses provedores como a OutSystems, Microsoft Power Apps, Bubble, SAP Appgyver, Flutterflow e Oracle APEX.

Cada uma dessas empresas possui sua própria abordagem e conjunto de recursos para facilitar o processo de desenvolvimento de aplicativos com baixa necessidade de codificação, permitindo que usuários com diferentes níveis de conhecimento técnico criem aplicações personalizadas, utilizando interfaces visuais e elementos pré-construídos.

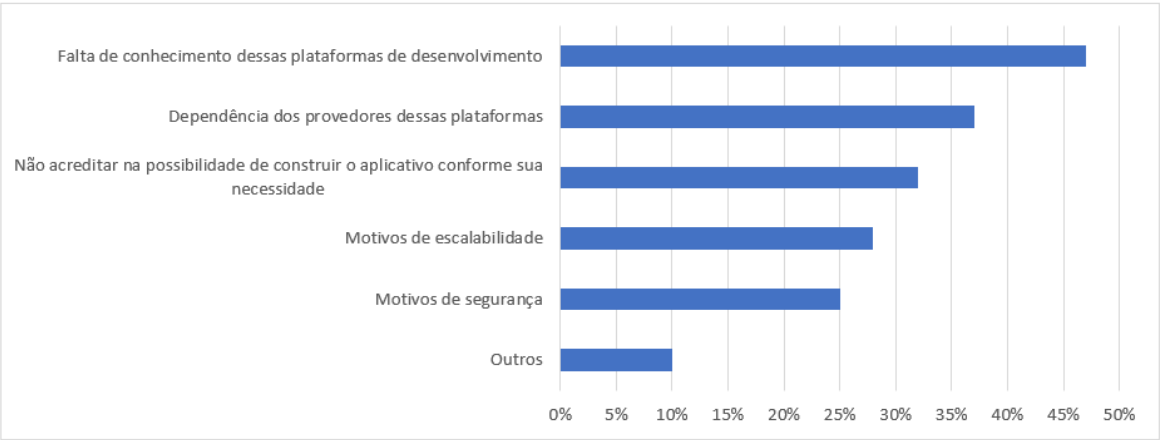
Uma pesquisa desenvolvida pela "*The State of Appliaction Development*" em março de 2009 ouviu 3300 profissionais da TI (Tecnologia da Informação) de diferentes continentes com o objetivo de coletar informações acerca das tecnologias de desenvolvimento. Nessa pesquisa, 34% dos entrevistados responderam que utilizam tecnologia de desenvolvimento "*low code*" e 9% afirmaram que sua organização estava prestes a começar a usar. Ademais, foi apontado os principais pontos positivos e negativos para a utilização de uma plataforma de desenvolvimento de baixo código mostrado nas Figuras 3 e 4 (OUTSYSTEMS, 2019).

Figura 3- Principais razões para a utilização de uma plataforma de baixo código



Fonte: (OUTSYSTEMS, 2019) traduzido pelo autor

Figura 4- Por que de algumas empresas não utilizarem plataformas de baixo código



Fonte: (OUTSYSTEMS, 2019) traduzido pelo autor

3 BUBBLE

Esse capítulo tem como objetivo apresentar a plataforma de desenvolvimento empregada no projeto, fornecendo uma visão geral dos conceitos abordados para facilitar a compreensão por parte do leitor. Ademais, serão apresentadas as principais funcionalidades, desenvolvimento de ações e banco de dados.

3.1 Visão geral

O Bubble caracteriza-se como uma plataforma de desenvolvimento de baixo código contendo comunidade ativa e crescente em todo Brasil. Essa plataforma tem como fundamento dar soluções eficazes com uma ampla quantidade de funcionalidades que permitem aos usuários desenvolverem aplicativos personalizados de forma rápida e intuitiva.

Para a construção de uma aplicação web, é importante levar em consideração dois principais aspectos:

1. *Front-end*: Abrange todos os elementos visuais que o usuário final visualiza e interage no aplicativo;
2. *Back-end*: Consiste na lógica por trás das interações dos elementos visuais e tudo aquilo relacionado à tratativa no banco de dados;

A plataforma tem um conceito semelhante à outras do mesmo nicho, a funcionalidade de “arrastar” e “soltar” empregada, facilita no manuseio de elementos gráficos no momento da construção do *front-end* de uma aplicação. Essa funcionalidade permite que o usuário selecione elementos pré-moldados, como botões, imagens, campo de texto e outros componentes e posicione na interface do aplicativo.

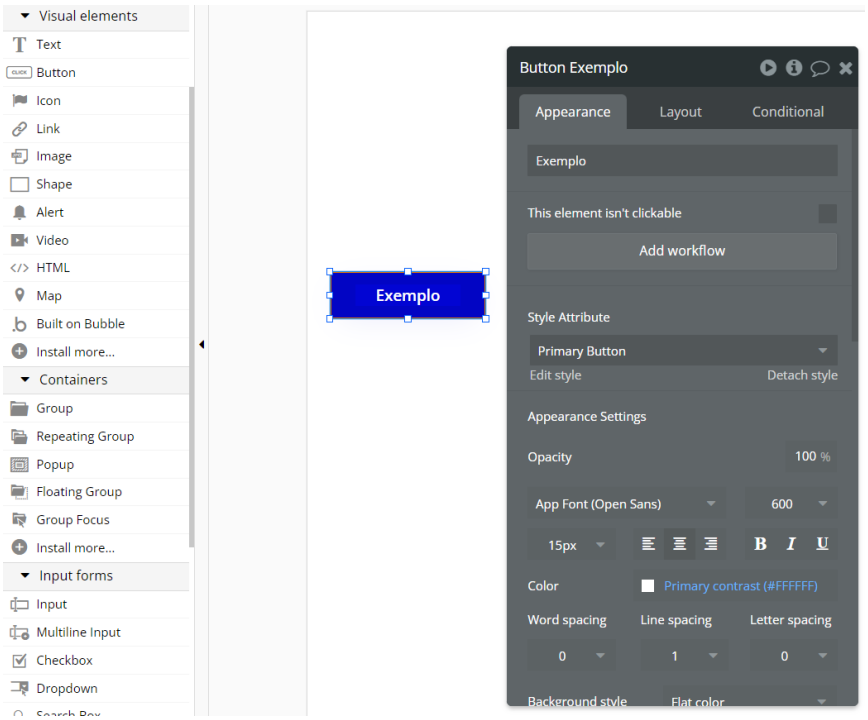
Esses elementos pré-moldados são componentes prontas que foram construídos por meio de linhas de código e são “encapsulados” dentro da plataforma de tal forma que facilite o processo de desenvolvimento por parte do usuário. Essa abordagem de “encapsulamento” também é empregada no *back-end*, simplificando a implementação da lógica e das ações na aplicação.

3.2 Front-end

A construção do *front-end* no Bubble consiste em arrastar elementos e blocos modelando a estrutura visual conforme a necessidade do usuário. Existem três principais tipos de elementos disponíveis:

1. Elementos visuais em geral: Abrange uma variedade de componentes como botões, caixas de texto, imagens, ícones e outros. Além disso, elementos personalizados pela comunidade também são caracterizados nessa categoria.
2. Containers: São utilizados para organizar e agrupar elementos visuais permitindo criar layouts e estruturas com um maior nível de complexidade dentro da aplicação.
3. Entrada de dados (*inputs*): São responsáveis por capturar informações inseridas pelo usuário, permitindo interações com o *back-end*.

Figura 5- Interface do Bubble para a construção do *front-end*



Fonte: Captura de tela da plataforma Bubble

Após a inclusão de um elemento gráfico, é possível alterar suas configurações de estilo e dimensionamento, personalizando de acordo com as preferências do usuário. Além disso, é possível adicionar lógica condicional aos elementos, definindo regras específicas para determinar ações de elemento que serão executadas quando certas condições forem atendidas, permitindo comportamentos de interações entre elementos visuais.

3.2.1 Grupo Repetidor

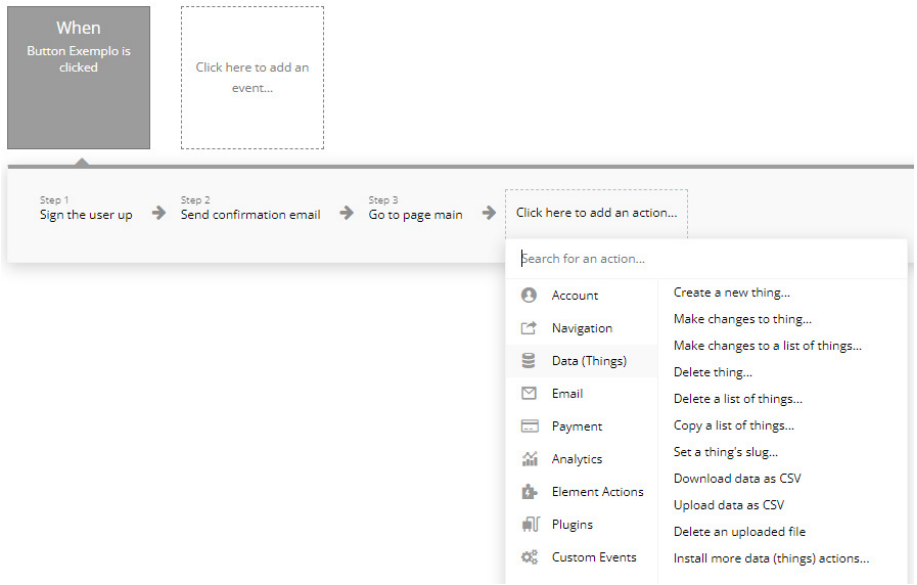
Um conceito de suma importância e que será abordado em diversas ocasiões do projeto será o de "*Repeating Group*" (Grupo Repetidor), no qual consiste em um container que possuirá uma fonte de dados vinculado a uma tabela do banco de dados. Com isso, será possível representar de forma dividida e organizada os dados referentes a cada linha dessa tabela, renderizando de forma repetida de acordo com o número de valores da mesma.

3.3 Back-end

3.3.1 Workflow

O *Workflow* consiste em um conjunto de ações ou um algoritmo que é executado em decorrência de uma situação específica definida pelo desenvolvedor, permitindo automatizar processos e controlar o fluxo de ações dentro de uma aplicação.

Figura 6- Interface do Bubble para a construção de *workflows*



Fonte: Captura de tela da plataforma Bubble

Essas ações são divididas em diferentes tipos conforme a interação necessária que a aplicação deverá ter com o usuário. Esses tipos podem ser:

1. Autenticação de usuário: Consiste em ações relacionadas ao gerenciamento de conta de usuários do sistema, permitindo registro, login, logout, recuperação de senha e outros.
2. Navegação: Gerenciamento de transição entre diferentes páginas e seções do aplicativo, permitindo que o usuário navegue pela interface e melhorando a sua experiência dentro da aplicação.
3. Dados: Consiste em ações responsáveis pelo gerenciamento do banco de dados. Essas ações envolvem manipulação e atualização de dados dentro da aplicação, incluindo a criação, edição, exclusão e busca de informações no banco de dados.
4. Email: Permite a possibilidade de envio automáticos de e-mails dentro da aplicação.
5. Pagamento: Essas ações estão relacionadas ao processamento de pagamentos dentro da aplicação, permitindo a integração com serviços externos e a realização de pagamentos de forma segura.
6. Analíticas: Ações utilizadas para a coleta e processamento de dados para análise e geração de indicadores.
7. Ações de elementos: Consiste em um conjunto de ações que irão interagir em elementos específicos dentro da aplicação.

3.3.2 Banco de dados

Diferentemente de outras plataformas de desenvolvimento de baixo código, o Bubble disponibiliza um banco de dados nativo integrado, fazendo com que não precise de uma conexão com um banco de dados externo e facilite o processo de manuseio de dados internos no momento da criação da aplicação. Ao utilizar esse banco de dados, o desenvolvedor tem a possibilidade de criar tabelas com variados tipos de dados e relacioná-las conforme a sua necessidade, possibilitando a criação de projetos com um maior nível de complexidade.

O método de relacionamento entre tabelas é semelhante a outros bancos de dados, utilizando chaves primárias e estrangeiras (chave de relacionamento). Cada linha de tabela, independentemente da tabela em questão, possui uma chave primária denominada "*Unique id*". O relacionamento entre tabelas é estabelecido ao vincular essa chave primária a outras tabelas, onde ela se tornará uma chave estrangeira. Esse mecanismo de relacionamento permite a conexão e a referência de dados entre diferentes tabelas do banco de dados.

O banco dados do Bubble está hospedado no servidor AWS (*Amazon Web Service*) com a utilização do serviço relacional Amazon RDS (*Relational Database Service*), garantindo a

confiabilidade, escalabilidade e segurança das informações com a utilização de criptografias no padrão AES-256, algoritmo que utiliza uma chave de 256 bits (BUBBLE, 2023).

3.3.3 Custom States

Consiste em uma variável de estado com um tipo definido que armazena valores conforme a necessidade da aplicação. Os *Custom States* (CS) fazem parte do *back-end* da aplicação e podem ser consideradas como um conjunto de dados armazenado em nuvem, ou de forma temporária, disponíveis somente na sessão do usuário que utilizará o sistema.

3.3.4 Option Sets

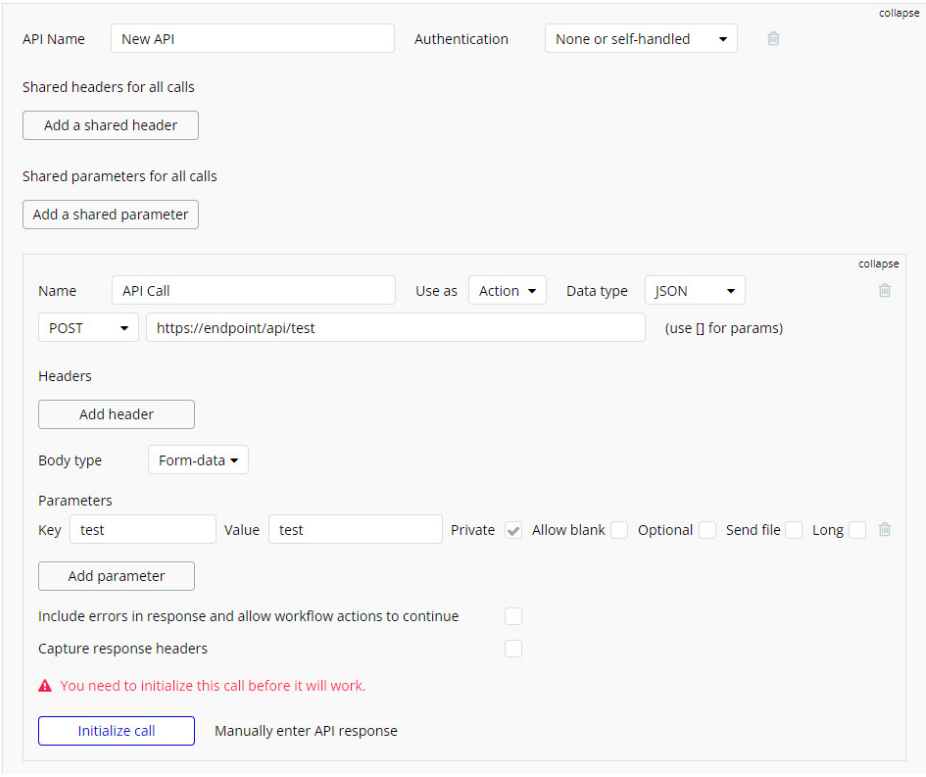
São conjuntos de dados que agrupam valores pré-definidos que não serão alterados no projeto. Esses grupos permitem que o usuário selecione entre opções específicas definidas pelo desenvolvedor durante o desenvolvimento do sistema. Por exemplo, pode-se criar um *option sets* do tipo “sexo” e definir os atributos como “masculino” e “feminino”. Esses atributos permaneceram constantes ao longo do projeto e não sofrerão alterações, garantindo uma padronização e consistência dos dados, tornando mais fácil a seleção de opções específicas

3.3.5 Integrações com terceiros

Uma Interface de programação de aplicação ou comumente chamada de API (*Application Programming Interface*), permite uma conexão ou comunicação entre sua aplicação desenvolvida com diferentes sistemas e serviços. Ademais, essa integração possibilita a interação e o compartilhamento de dados entre diferentes aplicações.

O Bubble oferece o “*API Connector*”, uma ferramenta nativa para desenvolvimento de chamados API que possibilita a configuração de requisições de diferentes tipos, simplificando a configuração de chamados para integração com aplicações e serviços externos.

Figura 7- Interface do Bubble para a configuração de requisições de chamado



Fonte: Captura de tela da plataforma Bubble

3.4 Plugins

São componentes adicionais desenvolvidos pela comunidade e disponibilizados para a utilização na aplicação, permitindo uma maior diversidade de funcionalidades que o Bubble não oferece de forma nativa. Os plugins podem ser disponibilizados de forma gratuita ou paga, dependendo da política adotada pelo desenvolvedor. A utilização de plugins amplia o leque de possibilidades como integrações com servidores externo de maneira facilitada, recursos avançados de design, manipulação de informações em bancos de dados, automações e outros.

3.5 Unidade de carregamento

Os "*Workload Units*" (Unidades de Carregamento) ou "Wu" consiste em uma métrica usada pelo Bubble para calcular o consumo de recursos da aplicação, tanto no *front-end* quanto no *back-end*. Cada ação na aplicação incluindo operações de banco de dados, fluxo de trabalho, interação na web e utilização de API externa são quantificadas nessa unidade de medida e a precificação é tomada com base nesse indicador.

3.6 Política de preços

O Bubble tem planos dos mais variados tipos, do gratuito até o customizável. No plano gratuito é possível utilizar 50kWu para a criação de uma aplicação com um limite de 200 linhas no banco de dados. O plano inicial (*Starter*) de 29,00$ mensais a aplicação pode consumir até 175kWu sem limitações no banco de dados. Para cada kWu adicionais, no plano "*Starter*" será consumido 0,30$ por kWu (BUBBLE, 2023).

4 DESENVOLVIMENTO DO SISTEMA

4.1 Considerações iniciais

O sistema foi dimensionado para trabalhar na coleta do registro de informações no que diz respeito a manutenção. Com isso, é essencial registrar dados relevantes, como o horário inicial e final, peças substituídas, equipe envolvida e local específico da máquina onde a manutenção foi realizada.

O sistema separa os ativos por tipos de máquina e centro de custo, de tal modo que facilite a análise de dados. Essa separação é feita por dentro do banco de dados do Bubble por meio de uma pré-configuração.

Para a padronização dos dados e uma análise mais precisa nas informações, é utilizado na aplicação uma metodologia de seções e subseções para a localização da manutenção. Nessa metodologia, a máquina é dividida em várias seções que correspondem às principais partes dela, conforme determinado pela equipe de gerenciamento do sistema. Além disso, cada seção pode ter subseções que representam seções dentro dessa principal. Por exemplo, considerando um computador, a CPU pode ser considerada uma seção e o conjunto de memória RAM uma subseção. Essa estrutura de seções e subseções permite uma organização hierárquica dos componentes da máquina, facilitando a identificação precisa do local da manutenção.

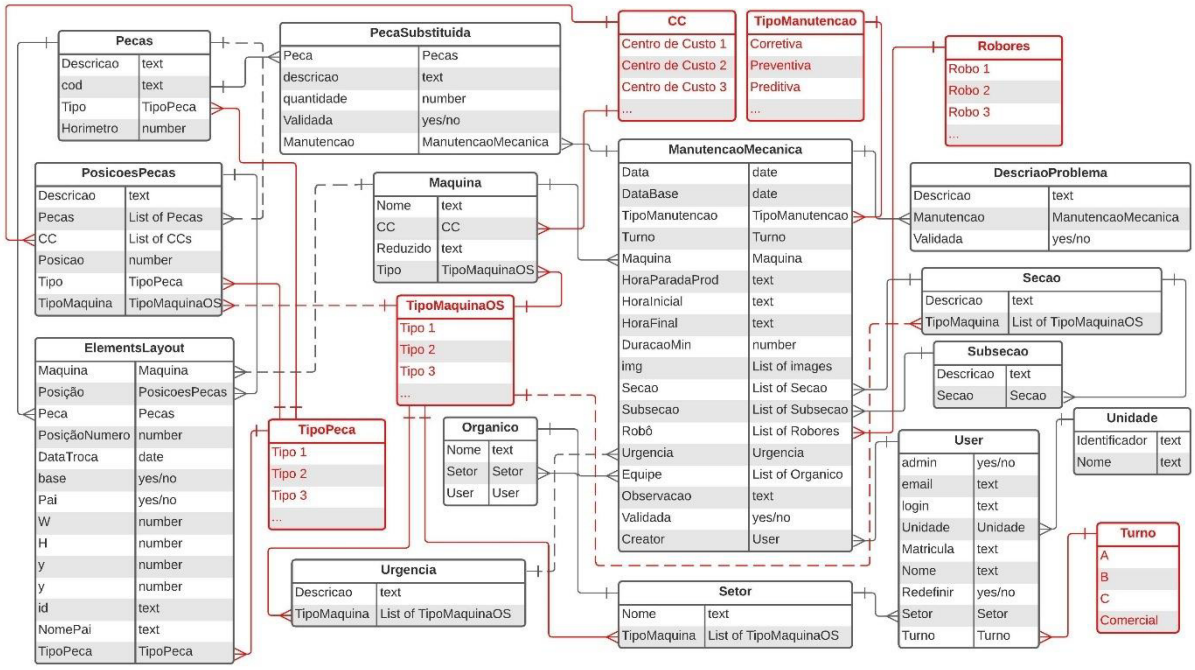
Será apresentado os principais *workflows* (algoritmos) implementados no projeto que impactam diretamente no funcionamento da aplicação, como a criação de registros de manutenção, manuseio desses dados em questão e no que diz respeito à elaboração de indicadores de manutenção.

4.2 Dimensionamento do banco de dados

O dimensionamento do banco de dados consiste em um dos principais requisitos para a inicialização desse projeto.

Ao criar uma nova coluna em uma tabela, o desenvolvedor tem a opção de escolher o tipo de dado dessa coluna, sendo possível selecionar outra tabela como tipo de dado. Na prática, ao vincular uma coluna a uma tabela, o Bubble realizará automaticamente essa conexão por meio do "*unique id*" dessa tabela. Dessa forma, a coluna criada estará implicitamente relacionada à tabela selecionada, permitindo a referência e a manipulação de dados entre as tabelas conforme a necessidade da aplicação.

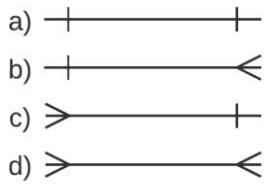
Figura 8- Diagrama de entidade e relacionamento



Fonte: Elaborado pelo autor

A Figura 8 apresenta uma modelagem de entidade e relacionamentos simples das tabelas que compõem o banco de dados estruturado no projeto. As partes vermelhas representam os *option sets* com atributos definidos no projeto conforme o seu tipo. As linhas interligadas representam a cardinalidade da relação em que:

Figura 9- Tipos de cardinalidade



Fonte: Elaborado pelo autor

a) Um para um;
b) Um para muitos;
c) Muitos para um;
d) Muitos para muitos;

4.3 Dimensionamento das telas

O Bubble disponibiliza recurso de responsividade que podem ser utilizados de acordo com as necessidades do projeto em desenvolvimento, permitindo a utilização da aplicação em diversos dispositivos independentemente do tamanho da tela.

No projeto, foi utilizado uma metodologia que separa duas versões do sistema, uma mobile e outra desktop. A versão mobile foi projetada e otimizada para funcionar de forma responsiva em dispositivos móveis, como smartphones e tablets, permitindo uma experiência de uso mais adequada com o intuito de facilitar o cadastro de registro de manutenção, visando a praticidade e agilidade por parte dos operadores. A versão desktop por sua vez, é destinada ao gerenciamento do sistema, análise de indicadores de manutenção e acompanhamento de peças conforme esse registro de informações.

No sistema, a versão desktop é destinada a um grupo específico de usuários que são responsáveis pela gestão do sistema e o mobile é projetado para atender a todos os usuários. Embora sejam destinados a diferentes grupos de usuários, ambos os sistemas utilizam as mesmas informações de credenciais de login, facilitando o uso diário do sistema no dia-dia, sendo diferenciado por diferentes tipos de permissões.

As duas versões da aplicação são dimensionadas no formato "*single page*", que consiste na colocação de todas as telas em uma única página. Essa abordagem oferece vantagens em termos de velocidade e desempenho, pois permite carregar todos os recursos da aplicação que será utilizado em uma única vez, possibilitando que o usuário da aplicação tenha uma

experiência mais fluida, com uma transição de telas mais suave e um tempo de resposta mais otimizado.

A dinamicidade das telas é dada por meio de *Custom State's*, que armazenam valores conforme a utilização e interação do usuário com o sistema. Com isso, é possível alterar elementos visuais, com exibição e ocultação de componentes, atualizando dados exibidos na tela.

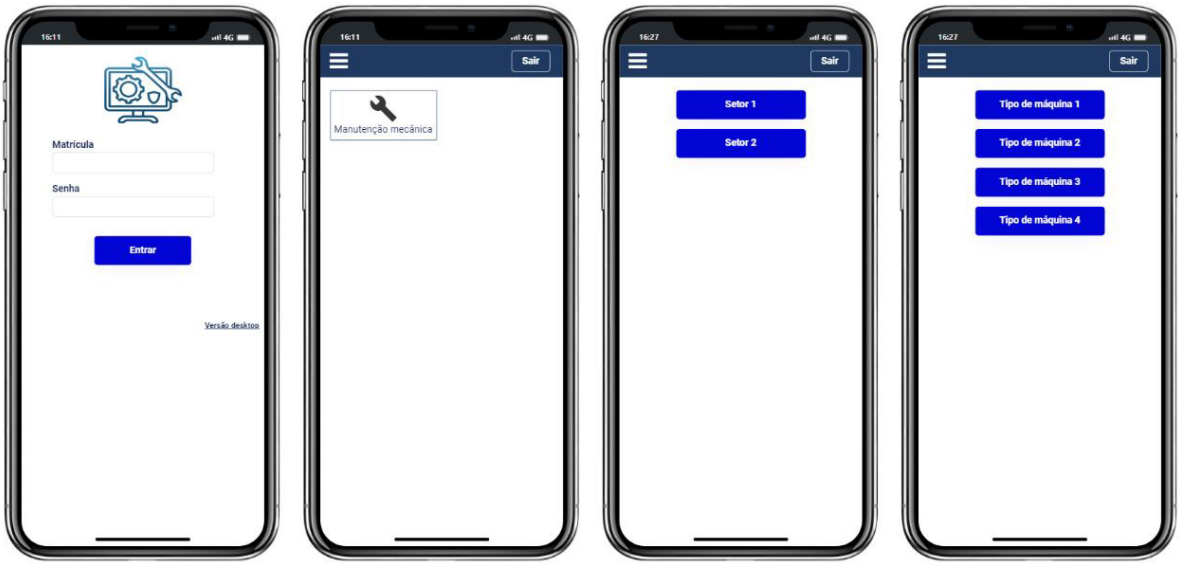
4.4 Versão mobile

Conforme mencionado anteriormente, a versão mobile do sistema tem como objetivo principal oferecer uma maior praticidade no manuseio do sistema. Essa versão foi projetada especificamente para permitir que os operadores registrem as ocorrências de manutenção conforme o setor que está alocado.

Antes de acessar a tela de registro de informações, o usuário terá que escolher qual setor e centro de custo irá querer realizar o registro da manutenção. Essa parte do sistema foi desenvolvida usando conceitos básicos de alocação de botões, e uma lógica simples de alterações de estados dos CS's para o condicionamento da visibilidade dos elementos.

O cadastro de usuários é realizado de forma exclusiva pelo desenvolvedor da aplicação diretamente no servidor do Bubble. Cada usuário terá suas informações cadastradas de forma particular, com os seus níveis de permissão de acesso, conforme a necessidade de uso.

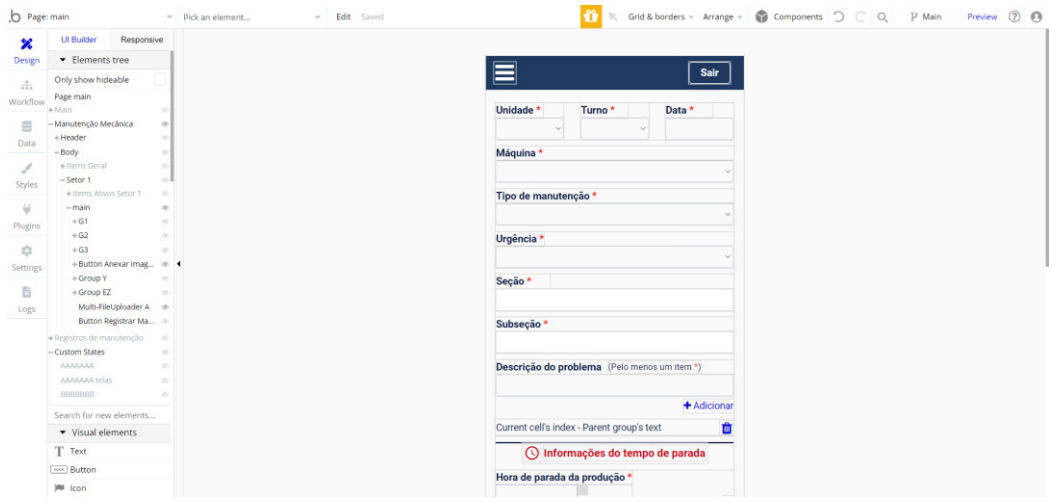
Figura 10- Telas iniciais da versão mobile



Fonte: Captura de tela da aplicação desenvolvida pelo autor

A divisão de ativos de forma sistematizada desempenha um papel fundamental na padronização de informações e em uma futura análise dessas informações. Portanto, durante o registro de manutenção, o usuário será solicitado para escolher o setor e o tipo de ativo em que a manutenção aconteceu, conforme a Figura 10, permitindo uma maior organização dos registros. Para desenvolver essa tela do sistema, foram utilizados conceitos básicos de alocação de botões e uma lógica simples, tornando o processo de escolha do setor e do tipo de ativo intuitivo e fácil de usar para o usuário.

Figura 11- Construção da tela de registro de manutenção



Fonte: Captura de tela da plataforma Bubble

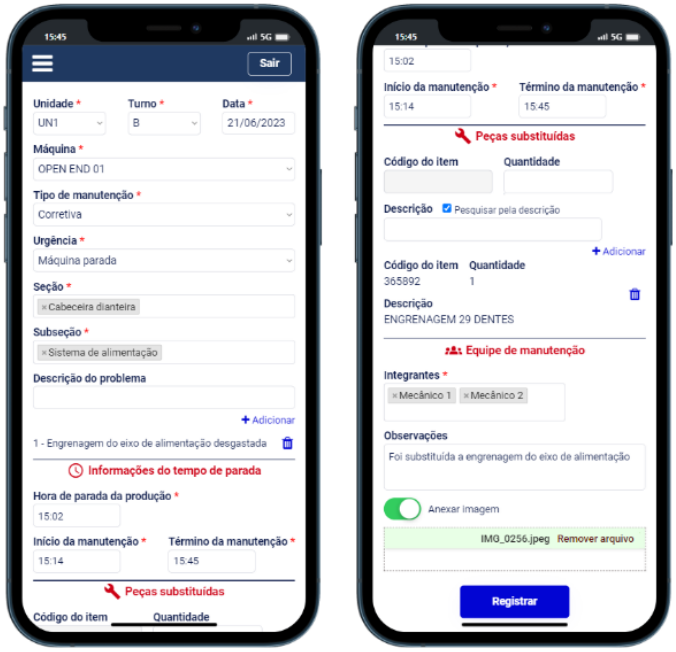
Em seguida, o usuário será direcionado para a tela principal de registro de informações, onde poderá inserir os principais dados da manutenção. Essa tela foi projetada para exigir o mínimo de informação possível por parte do usuário, visando a praticidade e agilidade da operação.

Para a construção de todas as telas, o principal recurso utilizado para agrupar os elementos visuais na tela foram os grupos do tipo container. Esse recurso permitiu que essas telas proporcionassem uma melhor experiência para o usuário, de forma intuitiva e amigável. Além disso, para cada botão de funcionalidade presente nas telas, como “adicionar” e “Registrar”, conforme a Figura 12, foi adicionado um *workflow* definindo as ações executadas ao clicar nesses botões.

As listas apresentadas na aplicação, com base nas informações cadastradas no banco de dados são representadas por um grupo repetidor. Esse grupo permite que cada linha da lista

corresponda a uma linha referente à tabela do banco de dados de acordo com os dados que o desenvolvedor queira expor.

Figura 12- Tela principal para o cadastro de registro de manutenção

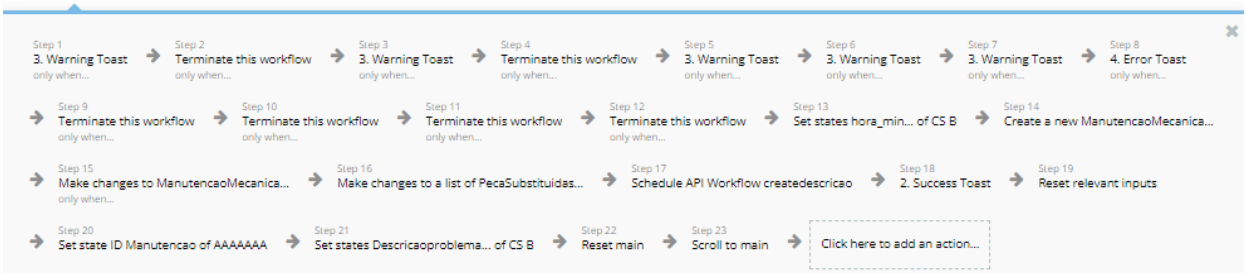


Fonte: Captura de tela da aplicação desenvolvida pelo autor

O campo de “Peças substituídas” pode ser integrado com o banco de dados do almoxarifado da empresa, facilitando a busca pela componente substituída na manutenção. Essa busca pode ser realizada tanto pelo código do item cadastrado, como pela sua descrição cadastrada no sistema.

O formulário de registro de informações contém campos obrigatórios, para que o usuário entre com os dados, e outros campos optativos, como é o caso do campo de observação e o de anexar imagens.

Figura 13- *Workflow* do botão “Registrar” do formulário de manutenção



Fonte: Captura de tela da plataforma Bubble

Após o preenchimento do formulário, o usuário deverá clicar no botão “Registrar”, e em seguida, será executado um *workflow* que irá validar as informações preenchidas e armazenar essas informações no banco de dados como um novo registro. Essas ações no workflow podem ser entendidas como:

1. *Warning Toast*, *Error Toast* e *Sucess Toast*: Ações do plugin instalado na aplicação (*myAlerts!*) que permite mostrar avisos na tela.
2. *Terminate this workflow*: Ação que permite interromper o fluxo de ações. Geralmente, essa função é usada em conjunto com uma condição para validar o momento da interrupção.
3. *Set states*: Permite alterar o valor de uma variável de estado (Custom State).
4. *Create a new*: Cria um registro no banco de dados de acordo com a tabela escolhida.
5. *Make changes to*: Permite fazer alterações em um registro de uma tabela do banco de dados.
6. *Make changes to a listo of*: Permite fazer alterações em uma lista de registros de uma tabela no banco de dados.
7. *Shedule API Workflow*: Programa um fluxo de ações que será executado diretamente no servidor do Bubble de forma assíncrona, sem interferir na interação do usuário.
8. *Reset relevants inputs*: Ação que reseta as caixas de texto do formulário.
9. *Reset*: Reseta os elementos de um container específico.
10. *Scroll to*: Ação que permite realocar a tela do usuário para uma determinada posição específica, tomando como base a posição de um container.

Os *steps* 1 ao 12 são ações com o objetivo de validar os valores inseridos pelo mecânico no formulário de preenchimento das informações. O *step* 13 irá realizar um cálculo da diferença de horários inicial e final de manutenção, em minutos, e armazenar na variável de estado “hora_min”, que está localizada no elemento “CS B”. O step 14 consistem na criação de uma nova linha na tabela “ManutencaoMecanica”, onde cada campo de entrada de dados do formulário é atribuído ao seu respectivo campo na coluna da tabela. O *step* 15 realiza uma validação de informações relacionada às colunas “Turno” e “DataBase” em que, caso a ocorrência da manutenção ocorra após a meia-noite (ainda no turno C e antes do turno A), a data base dessa ocorrência será do dia anterior. Os *steps* 16 e 17 têm como objetivo adicionar as peças substituídas à tabela “PecaSubstituida” e os registros de descrição do problema à tabela “DescricaoProblema”, respectivamente.

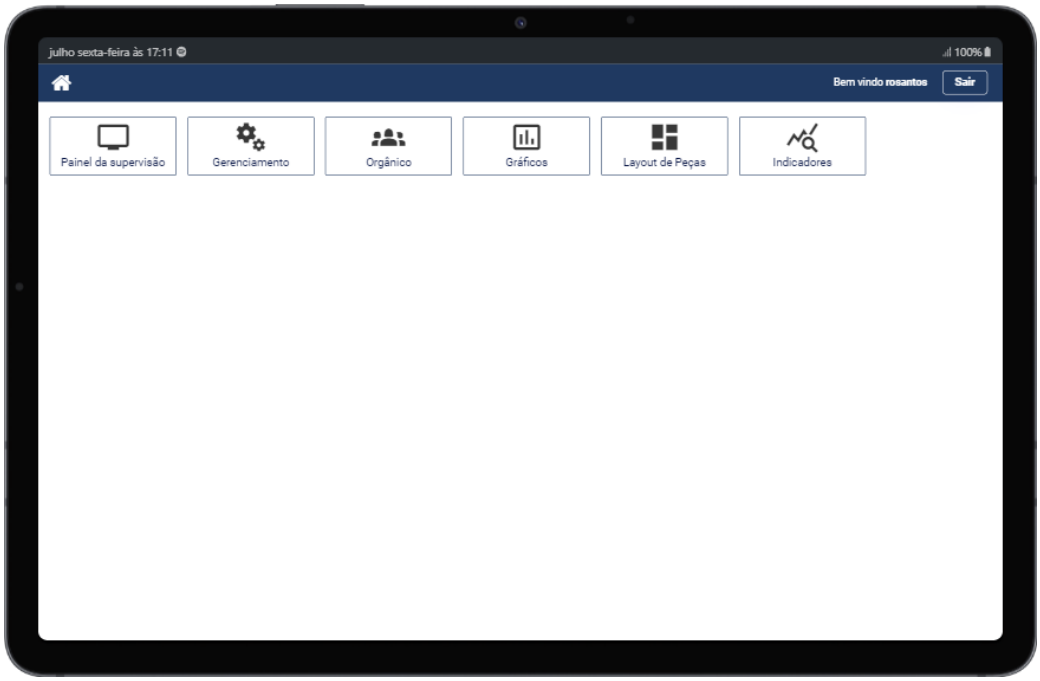
Para o *step* 18, com a utilização de uma API, foi configurado o envio pelo WhatsApp de forma automatizada das informações da manutenção após a realização do registro. Essa integração proporciona uma comunicação mais ágil e eficiente, reduzindo a necessidade de comunicação manual e aumentando a eficiência operacional por parte da equipe.

Por fim, os *steps* de 19 ao 24 irão finalizar as ações do *workflow*, mostrando um informativo na tela de que a manutenção foi registrada com sucesso e resetar os valores das variáveis de estado para um eventual próximo registro.

4.5 Versão desktop

Essa versão tem como proposito fazer com que os gestores e administradores do sistema possam consultar e validar as informações cadastradas pelos mantenedores. Além disso, tem a possibilidade de fazer um acompanhamento de peças e indicadores de manutenção no geral que irão impactar diretamente na tomada de decisão da equipe. A aplicação é dividida em alguns módulos que serão explicados posteriormente.

Figura 14- Tela inicial da versão desktop da aplicação

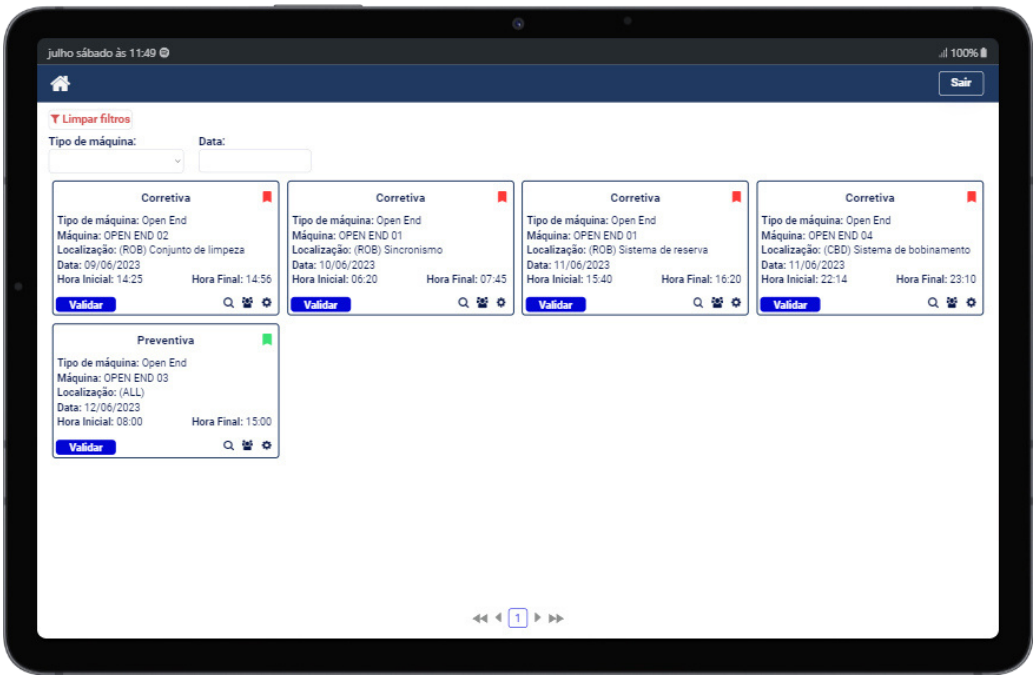


Fonte: Captura de tela da aplicação desenvolvida pelo autor

4.5.1 Painel da supervisão

Esse módulo consiste em duas opções de telas: “Validar Manutenções” e “Banco de dados”. Quando um registro de manutenção é cadastrado no banco de dados, ele é inserido como uma manutenção não validada, e o processo de gerenciamento e validação dessas manutenções ocorre nesta aba. A validação das manutenções é realizada pelo time de gestão responsável que irá verificar se todas as informações estão corretas e condizentes com a manutenção realizada. Caso haja alguma informação incorreta ou ausente, o gestor tem a possibilidade de fazer edições na manutenção para corrigir eventuais erros de cadastramento. A disposição das informações por manutenção é organizada em um grupo repetidor que exibe as informações cuja coluna “Validada” possui o valor “Não”.

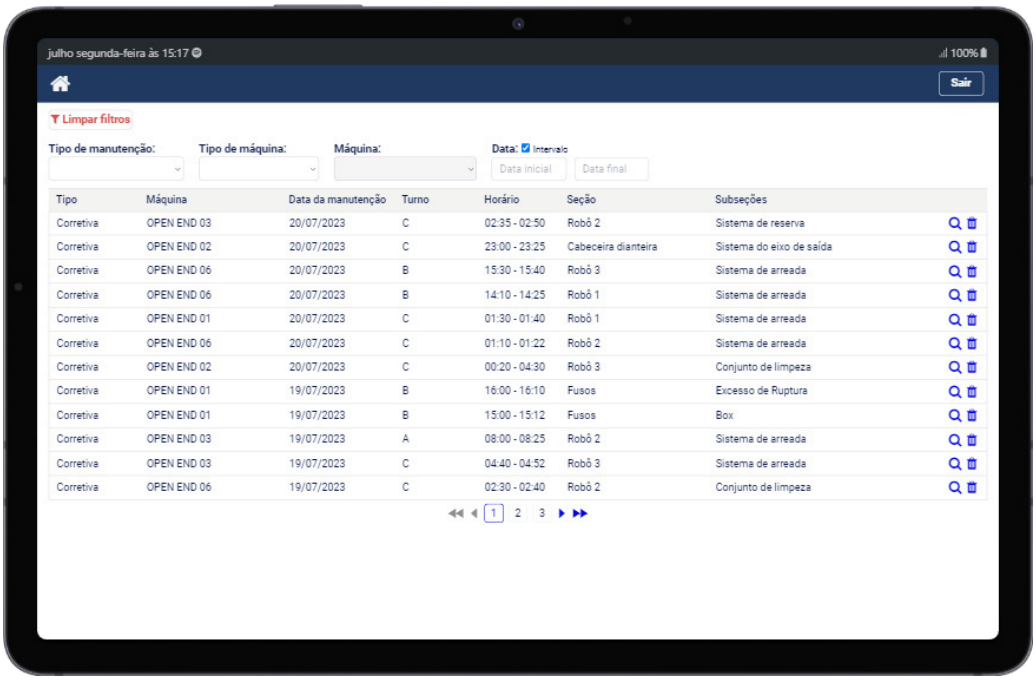
Figura 15- Tela gerencial do painel da supervisão (Validação de dados)



Fonte: Captura de tela da aplicação desenvolvida pelo autor

Após a validação dessa manutenção, esse registro de manutenção entra para o banco de dados das manutenções validadas, onde farão parte das informações que serão utilizadas para realizar as análises dos indicadores dentro do sistema, conforme mostrado na Figura 16.

Figura 16- Tela gerencial do painel da supervisão (Banco de dados)



Fonte: Captura de tela da aplicação desenvolvida pelo autor

Essa tela permite ao usuário realizar buscas específicas no banco de dados de manutenções validadas com base em diferentes critérios, aplicando diversos filtros relacionados por tipo de máquina, máquina ou datas definidas por uma data base ou intervalo de tempo.

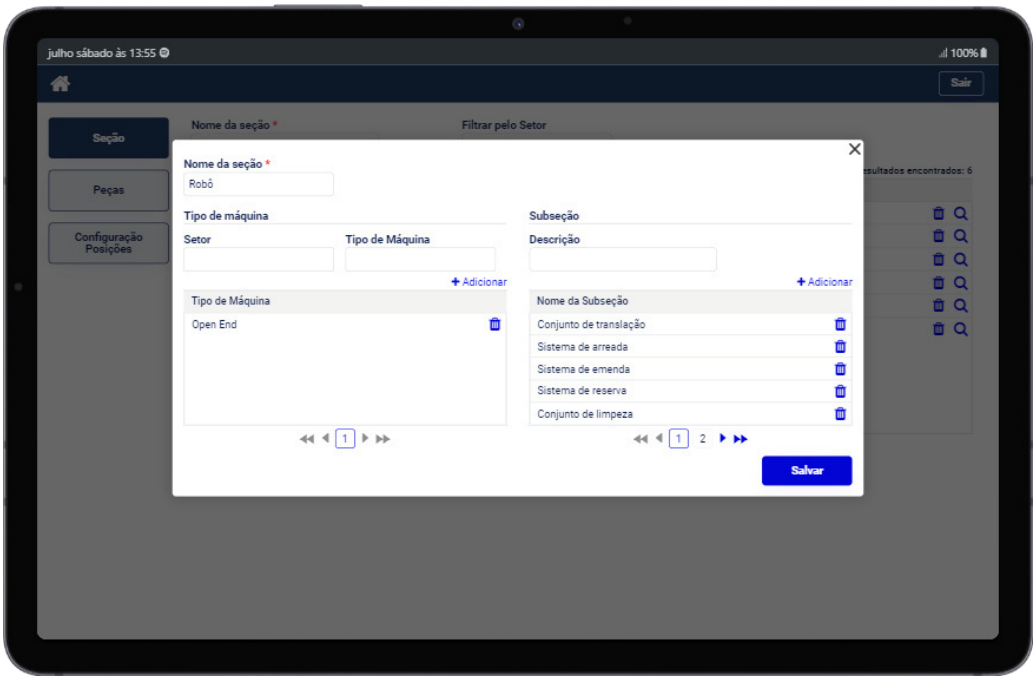
4.5.2 Orgânico

Esse módulo possibilita ao administrador do sistema fazer o gerenciamento da equipe de mecânicos disponíveis em seu quadro de colaboradores da empresa, atribuindo ao seu respectivo setor em que irá atua.

4.5.3 Gerenciamento

Tem como finalidade fazer o gerenciamento das informações que serão apresentadas na tela de registro de manutenção. Essa será uma das telas mais importantes antes da inicialização da coleta das informações, pois será onde os gestores irão realizar o "*setup*" das configurações do sistema. Esse módulo é divido em três partes fundamentais: "Seção", "Peças" e "Configuração de posições".

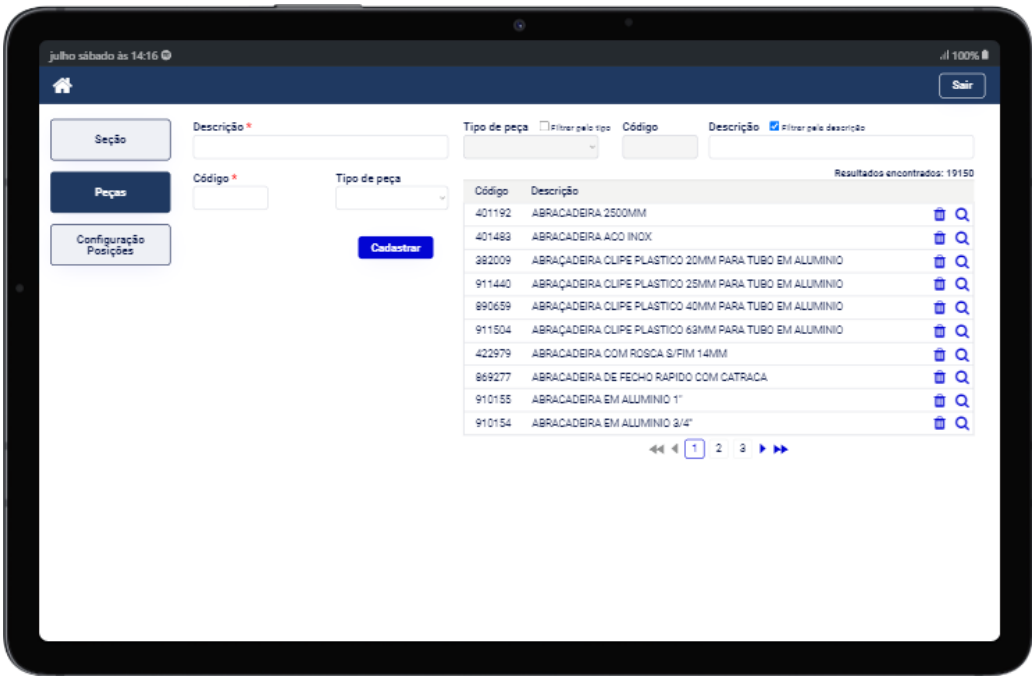
Figura 17- Tela de gerenciamento de seções (Seção “Robô” como exemplo de edição)



Fonte: Captura de tela da aplicação desenvolvida pelo autor

Na aba “Seção”, o administrador do sistema tem a possibilidade de cadastrar novas seções de acordo com o tipo de máquina e centro de custo, além de gerenciar as seções já existentes no sistema. Ao criar uma nova seção, o administrador deverá informar os detalhes dessa seção, como o nome, setor e tipo de máquina associada a ela. Em seguida, ele poderá cadastrar as subseções que compõem essa seção específica, hierarquizando os componentes da máquina de forma padronizada. Com a utilização de um container do tipo “*popup*”, é possível trazer os dados de um grupo repetidor, que correspondem a uma linha específica da tabela “Seção”, e mostrá-los em uma janela, como ilustrado na Figura 17.

Figura 18- Tela de gerenciamento de peças

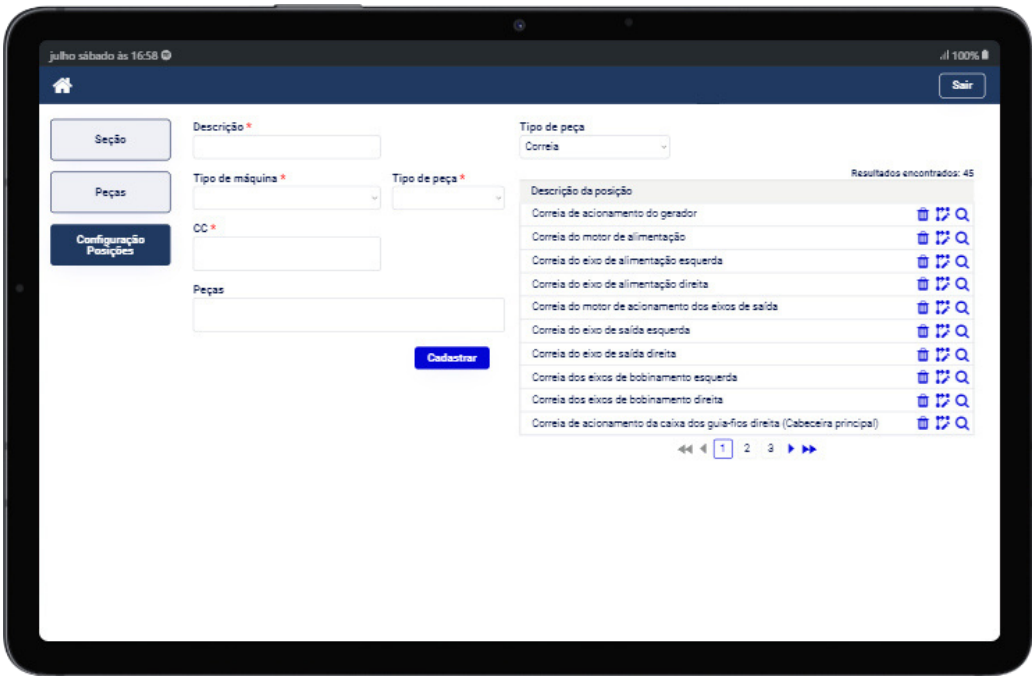


Fonte: Captura de tela da aplicação desenvolvida pelo autor

Para o gerenciamento de peças, os dados de entrada para o cadastro de um item são referentes à descrição da peça, código do item e tipo de peça conforme a Figura 18. O tipo de peça são valores pré-definidos pelo desenvolvedor, com base nas necessidades de análise de peças da empresa. Por exemplo, pode-se cadastrar peças do tipo “mancal”, “cilindro pneumático”, “correias” e outros, de acordo com as categorias consideradas pela empresa. Essa relação de tipo de peça está diretamente ligada a tela de “Layout de Peças” que será apresentada posteriormente.

Além disso, para cada peça que contém um determinado tipo, o administrador tem a opção de cadastrar o valor do seu horímetro correspondente a uma estimativa do tempo de vida útil daquela peça em questão.

Figura 19- Tela de gerenciamento de posições

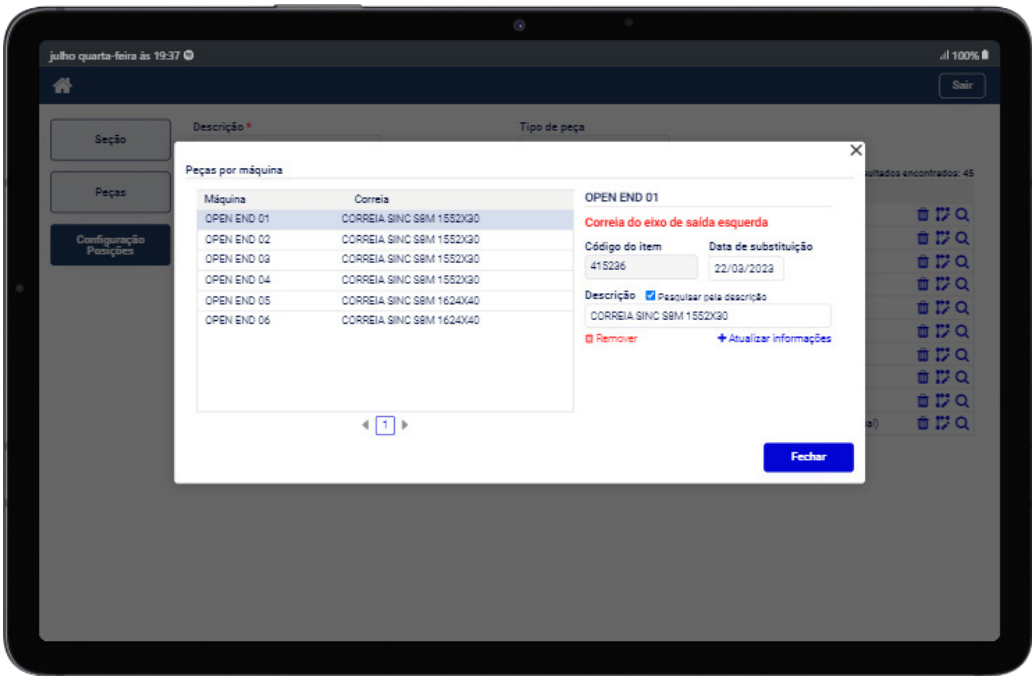


Fonte: Captura de tela da aplicação desenvolvida pelo autor

No módulo de gerenciamento de posições, as posições referem-se a locais específicos que abrigam peças de um determinado tipo cadastradas no sistema. Essas posições são divididas de acordo com a máquina, tipo de peça e o centro de custo em que cada posição criada para um determinado tipo de máquina, será representada por um retângulo no módulo de “Layout de Peças”.

O administrador do sistema terá a responsabilidade de cadastrar as posições de acordo com o tipo de peça. No sistema aplicado em testes, foram utilizadas apenas as peças do tipo “correia” para fins de demonstração. Cada posição cadastrada no sistema foi associada a uma posição específica de uma peça do tipo “correia” e armazenado na tabela “PosicaoPeca”. É importante destacar que pode haver mais de uma peça que seja compatível com uma mesma posição, desde que o centro de custo seja o mesmo.

Figura 20- Popup de gerenciamento de posição por máquina



Fonte: Captura de tela da aplicação desenvolvida pelo autor

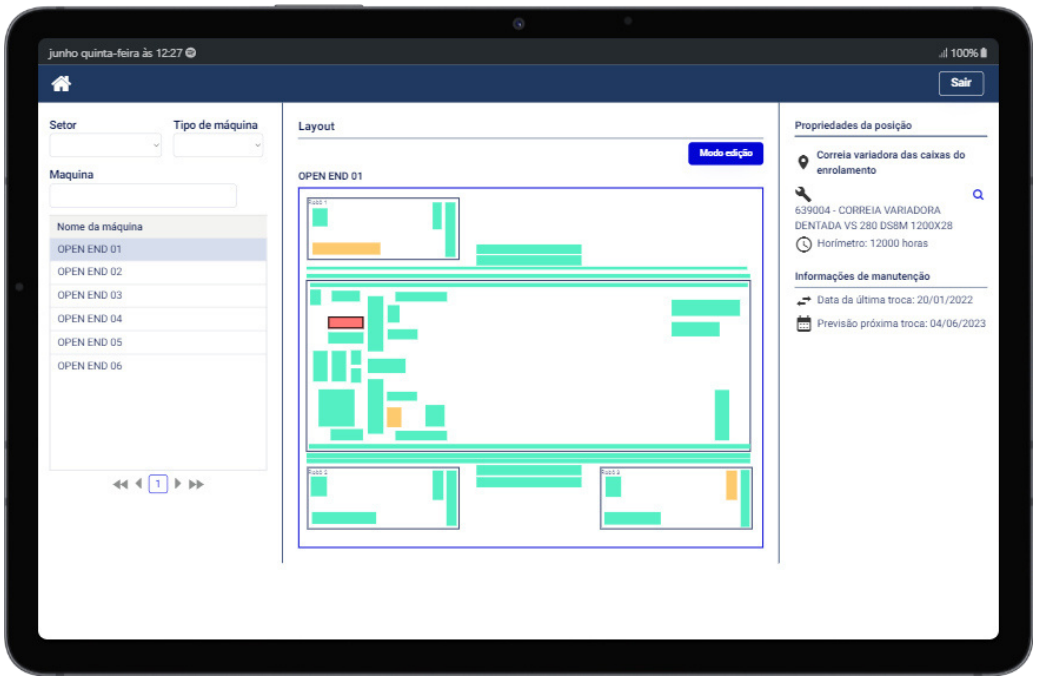
Escolhendo uma posição, o administrador do sistema tem a possibilidade de cadastrar a peça daquela posição conforme a sua data de substituição. Ao inserir a data de substituição da peça, o sistema calcula automaticamente, por meio de CS's, a data da próxima substituição com base no horímetro cadastrado da peça daquela determinada posição. Isso permite que o sistema faça um acompanhamento detalhado do tipo de peça em questão por máquina, auxiliando na gestão preventiva e planejamento de manutenções futuras.

4.5.4 Layout de Peças

Esse módulo tem como objetivo fornecer ao gestor uma visualização gráfica representativa da máquina, mostrando o conjunto de peças associadas a ela. Essa representação é composta por blocos, sendo que cada bloco representa uma peça específica da máquina, conforme o tipo de peça que queira realizar a análise.

Essa abordagem visual permite ao gestor realizar uma análise rápida e eficiente do estado das peças em relação à necessidade de substituição. Ao clicar em um bloco específico, será apresentado informações como a descrição do item que está naquela posição, data da última substituição e da próxima substituição conforme a Figura 21.

Figura 21- Layout de peças do tipo correia



Fonte: Captura de tela da aplicação desenvolvida pelo autor

A divisão dos blocos em cores tem como objetivo facilitar a interpretação e a leitura das informações relacionadas à data de troca de cada peça. Essa diferenciação por cores proporciona uma rápida identificação do estado de cada peça, simplificando o processo de análise e tomada de decisões por parte do gestor.

A cor verde é utilizada para indicar que a peça está dentro do período seguro de operação, ou seja, ainda não é necessário realizar a substituição, pois a peça ainda está em bom estado de funcionamento. A cor amarela é adotada para representar que a substituição da peça está se aproximando, indicando que não há a necessita de uma substituição imediata, mas que será necessário realizar a troca em um prazo de aproximadamente um mês. Por fim, a cor vermelha é utilizada para alertar que a data de substituição da peça já passou, indicando que a peça está em atraso para a troca e, portanto, é necessário tomar medidas corretivas imediatas para evitar problemas operacionais.

Além disso, é possível fazer uma edição de cada posição de peça representado no layout, customizando a representação das peças de acordo com o modelo real da máquina. Quando uma nova posição é criada no módulo de gerenciamento de posição, automaticamente é criado um bloco no layout de peças conforme aquele tipo de peça específico.

A customização da realocação de elementos na tela foi realizada através da utilização de plugins da comunidade. Como o Bubble não oferece uma funcionalidade nativa para essa

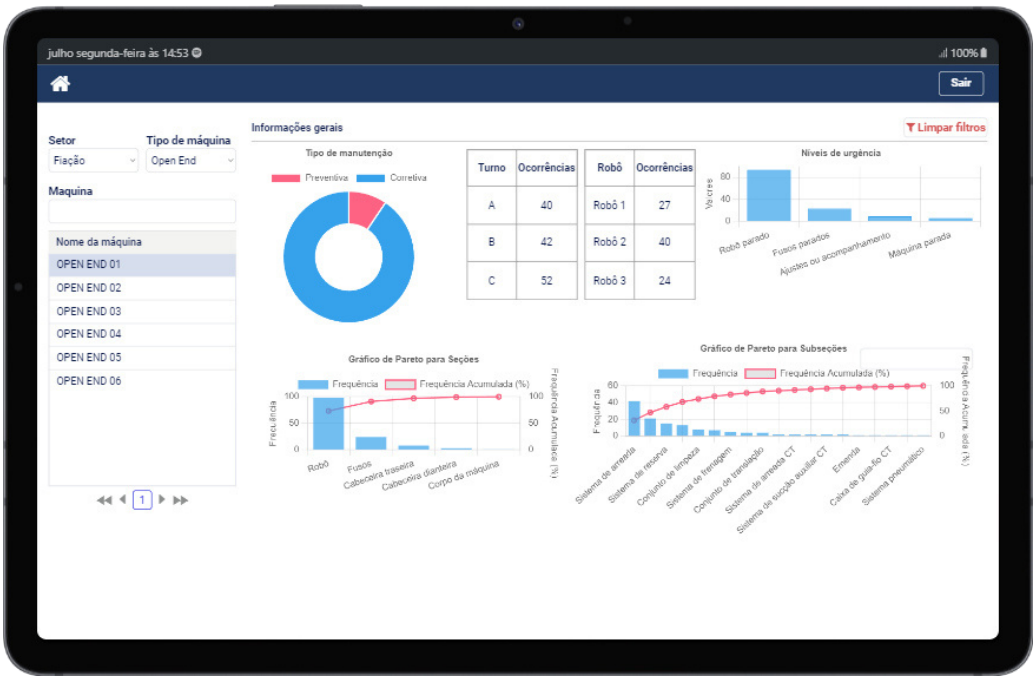
manipulação em tempo de execução, foi utilizado ferramentas de tal modo que possibilitasse essa funcionalidade. Por padrão, o carregamento dos elementos visuais na tela ocorre de acordo com suas localizações iniciais. No entanto, através do plugin, é possível buscar as informações de posicionamento e dimensões de cada elemento armazenadas na tabela "ElementsLayout" e, em seguida, carregar os elementos de acordo com suas configurações específicas.

4.5.5 Gráficos

O módulo "Gráficos" foi desenvolvido com o objetivo de fornecer ao gestor uma ferramenta para visualizar, em tempo real, o número de ocorrências de manutenções corretivas em diferentes tipos de análises. Nesse módulo, é possível acessar indicadores como o gráfico de Pareto das hierarquizações implementadas no sistema, como as seções e subseções conforme a Figura 22.

Para desenvolver esses gráficos, foi utilizado o elemento "HTML" disponibilizado pelo Bubble. Esse elemento permite a implementação de código HTML personalizado para customizar os elementos gráficos apresentados no módulo. Foi utilizado a biblioteca "Chart.js" para a customização dos gráficos.

Figura 22- Painel de gráficos

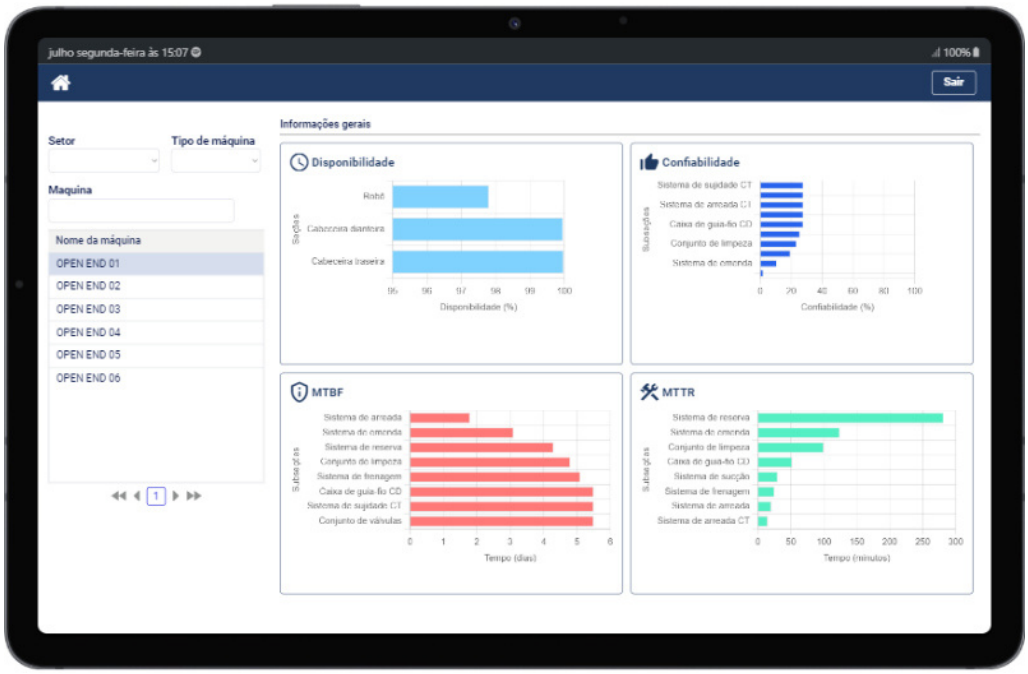


Fonte: Captura de tela da aplicação desenvolvida pelo autor

4.5.6 Indicadores

Esse módulo tem como principal objetivo trazer indicadores de manutenção que serão analisados pelo time de gestores da empresa. Todas as manutenções registradas e validadas no sistema iram ser computáveis e o sistema, de forma automatizada, irá realizar a série de cálculos tendo como resultado indicadores mostrado na Figura 23.

Figura 23- Painel de indicadores de manutenção



Fonte: Captura de tela da aplicação desenvolvida pelo autor

5 APLICAÇÃO

O sistema em questão foi aplicado em uma indústria têxtil de médio porte, no setor de fiação, utilizando somente um tipo de máquina específico chamado "*Open End*".

Como a empresa não tinha um sistema de manutenção interno ou terceirizado, devido ao alto custo de mercado desses CMMS, o gerenciamento de manutenção era feito de forma manual e escrito de tal forma que não possibilitava uma análise mais abrangente relacionado a elaboração de indicadores de manutenção.

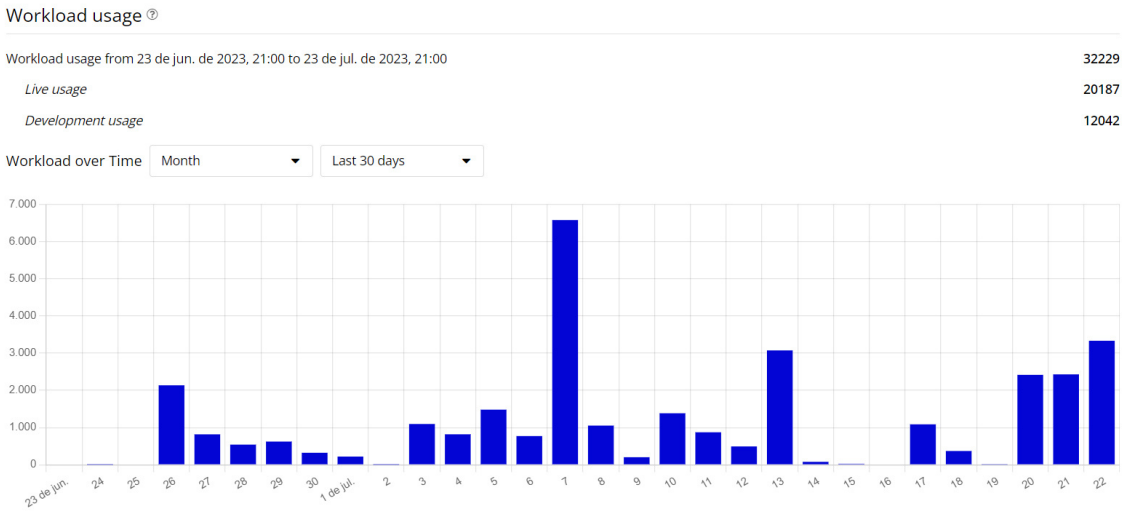
Para solucionar essa questão, o sistema desenvolvido foi implementado como uma solução para automatizar e otimizar o gerenciamento de manutenção. Como a aplicação ainda está em fase de validação na empresa, foi direcionado a um grupo de pessoas específicas, cadastrar todos os registros de manutenção voltadas aquele tipo de máquina.

Os registros da utilização do sistema apresentado nesse trabalho foram apresentados no capítulo de “Desenvolvimento do Sistema”.

5.1 Consumo da aplicação

Para a aplicação está em funcionamento, foi utilizado o plano “*starter*” do Bubble no qual possibilita a utilização de até 175kWu sem limitações de linhas no banco de dados. Esse planou supriu perfeitamente a demanda de utilização de workflows implementados no sistema. A utilização mensal é representada na Figura 24, onde a média de unidades de carregamentos fica em torno de 1074,3Wu/dia.

Figura 24- Consumo de unidades de carregamento



Fonte: Captura de tela da plataforma Bubble

6 CONCLUSÃO

No presente trabalho, apresentou-se o sistema desenvolvido usando uma ferramenta de desenvolvimento de baixo código como uma possível solução para empresas de pequeno e médio porte que não utilizam sistemas voltados ao gerenciamento de manutenção.

Com a utilização do sistema desenvolvido, a empresa pôde organizar seus dados de forma padronizada e hierarquizada de acordo com o tipo de máquina analisado. Outrossim, foi possível proporcionar ao time de gestores, indicadores para suas tomadas de decisões de tal modo que impactasse diretamente na produção e receita da empresa, visto que foi realizado um

estudo de aprimoramento de manutenções preventivas, atuando naquelas seções e subseções com uma maior incidência de ocorrências.

Um ponto importante a ser destacado é que o sistema foi desenvolvido utilizando apenas o plano pago “Starter” do Bubble e consumindo uma quantidade baixa de unidades de carregamento, o que resultou em um “Custo x Benefício” extremamente favorável para a empresa.

7 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

A contínua pesquisa e exploração de novas funcionalidades e recursos disponibilizados pelas plataformas de desenvolvimento de baixo código é de fundamental importância, pois essas tecnologias estão em constante evolução e inovação.

Existem diversas outras tecnologias de baixo código disponíveis no mercado que atendem o mesmo propósito de desenvolver ferramentas para a otimização de processos. Logo, é interessante explorar mais essas ferramentas existentes para que futuras soluções possam se embasar nessas pesquisas.

REFERÊNCIAS

COSTA, Elias Moura Júnior. **Sistema de manutenção**: Proposta de um modelo para empresa que não possua sistema integrado de manutenção. 1. ed. [*S. l.*]: Conhecimento Livre, 2019.

KARDEC, A; NASCIF, J. **Manutenção: função estratégica**. 3. ed. [S. l.]: Editora Qualitymark, 2009.

VIANA, Herbert Ricardo Garcia. **PCM: planejamento e controle da manutenção**. Rio de Janeiro: Quality, 2002.

FILHO, Gil Branco. **A Organização, o Planejamento e o Controle da manutenção**. Rio de Janeiro: Ciência Moderna Ltda, 2008.

GERHARDT, Tatiana Engel; SILVEIRA, Denise Tolfo. **Métodos de pesquisa**. 1. ed. [*S. l.*]: Editora eletrônica Luciane Delani, 2009.

XENOS, Harilaus G. **Gerenciando a manutenção produtiva**: O caminho para eliminar falhas nos equipamentos e aumentar a produtividade. [*S. l.*]: Editora DG, 1998.

OUTSYSTEMS. **The State of Application Development**: Is IT Ready for Disruption? Boston, MA, USA, 2019.

SANCHIS, Raquel; PERALES, Óscar García; FRAILE, Francisco; POLE, Raul. Low-Code as Enabler of Digital Transformation in Manufacturing Industry. **Applied Sciences**, [*S. l.*], p. 1-17, 18 dez. 2019.

BUBBLE. Bubble Docs. Disponível em: https://manual.bubble.io/. Acesso em: 20 mai. 2023.

APÊNDICE A – LISTA DE PLUGINS UTILIZADOS NO DESENVOLVIMENTO DA APLICAÇÃO

Air Date/Time Picker – Possibilita a criação de um input totalmente personalizável na tela que permite que o usuário entre com dados do tipo data ou horário. Disponível em: https://bubble.io/plugin/air-datetime-picker-1495642567089x595986733356023800.

CSS Loading Animations – Permite criar animações de carregamento estilizados em CSS na tela. Disponível em: https://bubble.io/plugin/css-loading-animations-1615402288950x110240248310005760.

Custom States Repeat – Permite criar uma lista de textos, números, datas ou endereços com valores repetidos. Disponível em: https://bubble.io/plugin/custom-states-repeat-1638556030475x573759375355412500.

Input Mask – Possibilita a criação de padrões de validação para inputs do tipo texto. Disponível em: https://bubble.io/plugin/input-mask-1609444246883x924984661248573400.

Moveable Elements – Possibilita ao usuário a funcionalidade de redimensionamento e realocação de elementos na tela. Disponível em: https://bubble.io/plugin/moveable-elements-1653809905398x530006895468740600.

MyAlerts! – Permite a criação de mensagens de aviso na tela, totalmente estilizadas pelo desenvolvedor. Disponível em: https://bubble.io/plugin/myalerts-1529535510382x166541069778419700.

Search & Autocorrect – Permite coletar valores instantaneamente de inputs do tipo texto e filtrar restrições com esse campo colunas de tabelas. Disponível em: https://bubble.io/plugin/search-autocorrect-1515542335452x628730204959539200.

Simple looper (workflow repeater) – Mecanismo que permite a repetição condicional de workflows. Disponível em: https://bubble.io/plugin/simple-looper-workflow-repeater-1618892957212x885363265747026000.