UNIVERSIDADE FEDERAL DO PARANÁ
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE ESPECIALIZAÇÃO EM ENGENHARIA AUTOMOTIVA

DESENVOLVIMENTO DE UMA FERRAMENTA GERENCIAL, UTILIZANDO BANCO DE DADOS ORIENTADO A GRAFOS

VINÍCIUS BATISTA ALVES SALGADO

Orientador: Luis Henrique Assumpção Lolis
Universidade Federal do Paraná – UFPR

CURITIBA
AGOSTO DE 2023

VINÍCIUS BATISTA ALVES SALGADO

DESENVOLVIMENTO DE UMA FERRAMENTA GERENCIAL, UTILIZANDO BANCO DE DADOS ORIENTADO A GRAFOS

Monografia apresentada à Banca Examinadora Trabalho de conclusão de curso para o curso de especialização em Engenharia Automotiva Setor de Tecnologia, pela Universidade Federal do Paraná, em cumprimento às exigências legais.

Orientador: Luis Henrique Assumpção Lolis
Universidade Federal do Paraná – UFPR

UNIVERSIDADE FEDERAL DO PARANÁ
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE ESPECIALIZAÇÃO EM ENGENHARIA AUTOMOTIVA
CURITIBA
AGOSTO DE 2023

VINÍCIUS BATISTA ALVES SALGADO

DESENVOLVIMENTO DE UMA FERRAMENTA GERENCIAL, UTILIZANDO BANCO DE DADOS ORIENTADO A GRAFOS

Monografia apresentada à Banca Examinadora Trabalho de conclusão de curso para o curso de especialização em Engenharia Automotiva pela Universidade Federal do Paraná, em cumprimento às exigências legais.

Trabalho aprovado. Curitiba, 11 de Agosto de 2023

Luis Henrique Assumpção Lolis
Orientador

Professor
Convidado 1

UNIVERSIDADE FEDERAL DO PARANÁ
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE ESPECIALIZAÇÃO EM ENGENHARIA AUTOMOTIVA
CURITIBA
AGOSTO DE 2023

Agradecimentos

Gostaria de expressar meu mais profundo agradecimento a todos que desempenharam um papel vital na conclusão desta tese.

Em primeiro lugar, agradeço ao professor Luis Henrique Assumpção Lolis por sua atenção, suporte acadêmico e compreensão. Seus ensinamentos e experiência foram fundamentais para moldar esta pesquisa.

Gostaria de agradecer também Mitch Boyer e Jacob Igney por todo o apoio coorporativo. Vocês me proporcionaram um ambiente de aprendizado prático, desafiador e estimulante, e a oportunidade de aplicar meus conhecimentos acadêmicos a situações do mundo real.

À minha família, à Cecilia e amigos, obrigado pelo seu amor incondicional, paciência e apoio constante ao longo desta jornada. Vocês foram a minha força e a minha inspiração.

Agradeço à UFPR, pela disponibilização de todas as aulas e pela oportunidade de estar cursando esta pós-gradução.

Este trabalho é o resultado da contribuição de muitas pessoas. A todos vocês que, direta ou indiretamente, fizeram parte desta jornada, meu sincero obrigado.

Resumo

Esta tese aborda a aplicação de técnicas de gerenciamento de processos e análise de risco, como FMEA (Análise dos Modos de Falha e seus Efeitos), para a otimização do processo de calibração de motores.

O estudo concentra-se na comparação da eficácia entre a gestão de processos usando planilhas Excel tradicionais e um software de grafos desenvolvido especificamente para essa finalidade. O software de grafos é apresentado como uma ferramenta mais eficiente devido à sua capacidade de visualização mais intuitiva, permitindo uma maior compreensão e controle dos processos em execução.

Os resultados apresentados na monografia indicam que o software de grafos tem um impacto positivo significativo na otimização de processos. Os indicadores de FMEA apresentaram melhorias, demonstrando uma eficácia aprimorada em termos de desempenho de equipamentos, diminuição de defeitos e aumento da eficiência.

Além disso, a tese mostra que o software permite uma gestão de processos mais eficiente, com um planejamento melhorado, capacidade de resposta mais rápida a problemas e melhor alocação de recursos.

Por último, ressalta-se a importância da integração da equipe na utilização da nova ferramenta, proporcionando uma compreensão compartilhada e uma comunicação mais eficaz entre os gestores e o time de engenharia.

Em resumo, a monografia defende a adoção do software de grafos como uma alternativa superior às técnicas de gestão de processos tradicionais para a otimização do processo de calibração de motores.

Palavras-chave: grafos. javascript. web. algoritmo. FMEA.

Abstract

This thesis addresses the application of process management and risk analysis techniques, such as FMEA (Failure Mode and Effects Analysis), for the optimization of the engine calibration process.

The study focuses on comparing the effectiveness of process management using traditional Excel spreadsheets and a graph software specifically developed for this purpose. The graph software is presented as a more efficient tool due to its more intuitive visualization capability, allowing for a greater understanding and control of the processes in operation.

The results presented in the thesis indicate that the graph software has a significant positive impact on process optimization. The FMEA indicators showed improvements, demonstrating an enhanced effectiveness in terms of equipment performance, defect reduction, and efficiency increase.

In addition, the thesis shows that the software allows for more efficient process management, with improved planning, quicker response to problems, and better resource allocation.

Lastly, the importance of team integration in the use of the new tool is emphasized, providing a shared understanding and more effective communication between managers and the engineering team.

In summary, the thesis advocates for the adoption of graph software as a superior alternative to traditional process management techniques for the optimization of the engine calibration process.

Keywords: graphs. web. algorithm. javascript. FMEA.

Lista de Figuras

Figura 1 – Exemplo de projeto "linear", em que nenhuma tarefa é executada de forma simultânea . . . 3
Figura 2 – Gráfico de eficência e *Goal Posts* . . . 3
Figura 3 – Grafo Completo . . . 10
Figura 4 – Gráfico de gantt, e detalhamento do caminho crítico de projeto exemplo 14
Figura 5 – Exemplo de grafo plotado com a ajuda da biblioteca Force Graph 2D 21
Figura 6 – Exemplo de gráfico plotado com a ajuda da biblioteca Echarts . . . 22
Figura 7 – Arquitetura da ferramenta . . . 23
Figura 8 – Tela principal . . . 30
Figura 9 – Resultado da visualização no modo ordenado . . . 32
Figura 10 – Resultado da visualização no modo não ordenado . . . 32
Figura 11 – Gráfico de gantt, gerado pelo do algoritmo CPM a partir da busca executada pelo usuário . . . 33
Figura 12 – Gráfico de barras indicando o tempo previsto em que cada um dos recursos estará em uso durante o projeto . . . 34
Figura 13 – Botão *Speed Dial* e todas suas opções . . . 35
Figura 14 – Ilustração gráfica de todo o processo de calibração . . . 36
Figura 15 – Tela de configuração dos recursos disponíveis . . . 37
Figura 16 – Tela de configuração das datas alvo para os *Quality Gates* . . . 37
Figura 17 – Tela de seleção do motor desejado . . . 38
Figura 18 – Visão ordenada dos processos de calibração do motor selecionado . . . 38
Figura 19 – Visualização dos componentes e pontos de calibração do motor selecionado . . . 39
Figura 20 – Tela para busca de dados no servidor . . . 39

Lista de Tabelas

Tabela 1 – Matriz de escolha de linguagem/framework para visualização e integração com Neo4j. 26
Tabela 2 – Tabela FMEA: Análise de riscos utilizando a tabela em excel para controle do processo 41
Tabela 3 – Tabela FMEA: Análise de riscos utilizando a aplicação web para controle de processos 42

Lista de Abreviaturas e Siglas

OEE	Overall Equipment Effectiveness
FTY	First Time Yield
FMEA	Análise de Modo e Efeitos de Falha
RPN	Número de Prioridade de Risco
CPM	Critical Path Method
ES	Early Start
EF	Early Finish
LS	Late Start
LF	Late Finish

Sumário

1 – Introdução **1**
1.1 Contextualização 2
1.2 Objetivos 5
1.3 Organização do trabalho 5

2 – Fundamentação Teórica **7**
2.1 Calibração de motores 7
2.2 Teoria de Grafos 9
2.2.1 Definição de Grafos 10
2.2.2 Utilização de Grafos 11
2.2.3 Métodos em Teoria de Grafos 11
2.2.4 Algoritmos de grafos 12
2.2.5 O método CPM 13
2.2.6 Requisitos do Método CPM 15
2.2.7 Grafos em Gestão de Processos e Controle de Qualidade 16
2.2.8 Grafos em Calibração de Motores 16
2.3 Neo4j 17
2.3.1 Sintaxe e exemplos de consultas Cypher 17
2.4 JavaScript e Node.js 20
2.4.1 JavaScript 20
2.4.2 Node.js 20
2.4.3 ForceGraph2D 21
2.4.4 Echarts 22

3 – Desenvolvimento **23**
3.1 Identificação de especialistas 23
3.2 Coleta de dados 24
3.3 Modelagem do banco 25
3.4 Implementação da Visualização do Grafo em uma Ferramenta Web . . . 25
3.5 Dagre 26
3.6 Análise de Impacto e Melhoria de Processos 27
3.6.1 Análise de FMEA 27
3.6.2 Outras métricas e Indicadores-Chave de Desempenho (KPI - *Key Performance Indicators* 28

4 – Apresentação do protótipo **30**

4.1 Descrição da ferramenta 30
4.1.1 Tela principal 30
4.1.1.1 Janela de vizualização 31
4.1.1.2 *Speed Dial* 33
4.2 Aplicação das técnicas de gestão 36
4.2.1 *Failure Mode and Effect Analysis* - FMEA 37
4.2.2 OEE: Aumentar a Eficiência Global dos Equipamentos 43
4.2.3 Tempo de Ciclo: Reduzir a Duração do Processo 43
4.2.4 FTY: Aumentar a Eficiência do Processo 43

5 – Conclusão 44
5.1 Trabalhos futuros 44

Referências 46

1 Introdução

A calibração de motores é um processo essencial na indústria automobilística, responsável por otimizar o desempenho, eficiência energética e emissões de poluentes (HEYWOOD, 1988). Com o crescente foco em regulamentações ambientais mais rígidas, como o Euro 6 da União Europeia (COMMISSION, 2018) e os esforços globais para reduzir as emissões de gases de efeito estufa (AGENCY, 2020), a calibração de motores torna-se cada vez mais importante para atender a essas demandas.

Neste contexto, a aplicação da teoria dos grafos e da análise do Caminho Crítico (CPM) pode oferecer benefícios significativos ao processo de calibração de motores, permitindo uma gestão mais eficiente dos processos e um controle de qualidade aprimorado. A teoria dos grafos, por exemplo, pode ser utilizada para modelar a complexa interação entre diferentes variáveis do processo de calibração, enquanto a análise CPM pode ajudar a identificar as etapas críticas no processo e otimizá-las (KERZNER, 2017a).

A investigação e aplicação propostas neste estudo emergiram do envolvimento direto com os desafios e oportunidades que a empresa enfrenta diariamente. Deste modo, este estudo não se trata de um exercício teórico descontextualizado, mas de um esforço orientado à aplicação prática na busca por soluções e melhorias para a calibração de motores na indústria automotiva. A escolha do tópico, a metodologia adotada e a implementação proposta foram todas guiadas pela experiência prática e pela necessidade de contribuir para o aprimoramento dos processos e qualidade dentro da organização.

É importante esclarecer que todos os exemplos de código apresentados nesta tese são fornecidos apenas para fins demonstrativos, a fim de ilustrar como as bibliotecas e algoritmos mencionados podem ser utilizados. Devido a requisitos de segredo industrial, os códigos específicos utilizados na implementação da ferramenta de visualização do grafo na empresa multinacional do setor automotivo para a qual este trabalho foi desenvolvido não podem ser divulgados. Dessa forma, os códigos apresentados aqui são simplificados e podem não representar a complexidade e a especificidade dos códigos utilizados na prática.

A motivação para a realização deste trabalho vem da percepção de que a gestão de processos e o controle de qualidade são áreas-chave para a eficiência operacional e competitividade no setor automotivo. O setor tem testemunhado uma crescente complexidade em suas operações, impulsionada pelo avanço da tecnologia, as expectativas crescentes dos clientes e as demandas regulatórias cada vez mais rigorosas. A utilização de grafos para mapear e entender o processo de calibração de motores é um passo

importante para enfrentar esses desafios. Portanto, explorar como a integração da teoria dos grafos e da análise CPM no processo de calibração de motores pode melhorar o desempenho, a eficiência energética e a conformidade com as regulamentações ambientais. Além disso, este estudo busca desenvolver uma ferramenta baseada na Web em JavaScript que utilize o banco de dados Neo4j para realizar consultas e análises, facilitando a aplicação prática dessas teorias no contexto da calibração de motores.

A inovação na área de calibração de motores é crucial, uma vez que a indústria automobilística enfrenta desafios crescentes devido às mudanças climáticas, à poluição do ar e às pressões regulatórias (FENG et al., 2018). Ao aplicar técnicas avançadas de gestão de processos e controle de qualidade, este estudo tem o potencial de oferecer insights valiosos para a indústria e contribuir para o desenvolvimento de tecnologias automotivas mais sustentáveis.

1.1 Contextualização

Antes da implantação desta nova abordagem, a empresa enfrentava vários desafios na gestão e planejamento do processo de calibração de motores. A complexidade dos componentes e a interdependência das partes envolvidas tornavam a tarefa de mapear, visualizar e analisar os processos um esforço significativo. Com o tempo, as alterações de hardware nos motores poderiam resultar em uma ampla gama de impactos na calibração e esses efeitos eram muitas vezes difíceis de antecipar e de medir. A falta de uma visão holística e integrada desses processos levava a problemas como ineficiências na produção, custos elevados e uma maior probabilidade de erros. Ao longo deste capítulo, serão abordadas os antigos procedimentos envolvidos no processo de calibração e suas possíveis fragilidades, a fim de contextualizar o leitor e facilitar a comparação entre os métodos.

A abordagem anterior para o planejamento e gestão de um projeto de calibração empregava uma planilha do Excel como recurso primordial. Nesta estrutura, todas as atividades, incluindo os testes (*Test Processes*) e processamento de dados (*Data Processes*), eram apresentados em uma sequência linear, como em 1.

O arranjo permitia a estimativa rudimentar da duração necessária para cada projeto. Cada uma dessas planilhas utilizavam linhas de referência que traçavam a área de interesse da eficiência de execução dos projetos, os chamados *Goal Posts*, Figura 2. Estas linhas indicavam uma região especificada pela gerência em que um projeto deveria ser mantido, geralmente variando entre 70% e 85% de eficiência. Sendo, eficiência definida neste contexto como:

$$E_f = \frac{(T_e - T_o)}{T_t} \tag{1}$$

	Período de execução					
	1	2	3	4	5	6
Tarefa 1	■					
Tarefa 2		■				
Tarefa 3			■			
Tarefa 4				■		
Tarefa 5					■	
Tarefa 6						■

Figura 1 – Exemplo de projeto "linear", em que nenhuma tarefa é executada de forma simultânea

Onde:

- T_e é o tempo real gasto na execução de tarefas,
- T_o é o tempo em que certa célula de dinamômetro permanesce ociosa por manutenção, defeito, troca de motores ou outros motivos,
- T_t são todos os turnos de trabalho somados até a presente data,

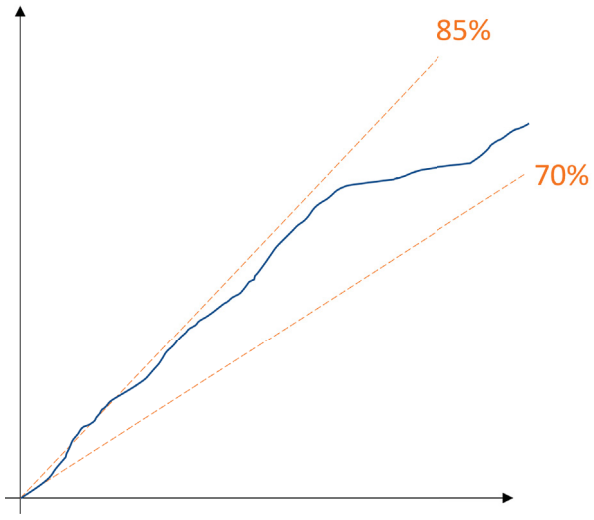

Figura 2 – Gráfico de eficência e *Goal Posts*

Esta forma de controle apresentava algumas dificuldades quanto ao controle das tarefas que deveriam ser executadas a cada momento.

1. Atrasos: Quando um projeto estava atrasado, ou com um baixo valor percentual de eficiência, a gerência tendia a alocar mais recursos para este projeto, como

utilizar duas ou mais células de dinamômetro para execução dos testes. Isso, por sua vez, se torna problemático pois eleva os custos e não se sabia ao certo o real efeito e o ponto em que o rendimento tornava-se decrescente.

2. Tempo ocioso ou *Flat lines*: Este é o tempo em que uma célula do dinamômetro ficava ociosa diminuindo consideravelmente a eficiência de execução da calibração, aguardando pelo processamento de dados coletados nos testes anteriores. Por diversas vezes, se tornava difícil identificar quais processamentos já poderiam ser executados em paralelo à execução de outros testes. Assim, identificava-se de forma tardia que um dado não havia sido processado, consequentemente, atrasando testes que necessitavam deste.

3. Paralelização: a vizualição no formato de lista, tornava difícil a vizualização de possíveis caminhos em paralelo, dependendendo integralmente da experiência dos gestores e engenheiros envolvidos no circuito. E mesmo sabendo-se que certo caminho poderia ser feito em paralelo, era necessária a criação de diversas abas em uma mesma planilha, aumentando a complexidade e dificuldade de gestão do documento.

A adaptabilidade pode ser entendida como a habilidade de um sistema ou método se ajustar de forma eficiente e eficaz a novas condições, variáveis ou requisitos. Isso implica que um sistema adaptativo tem a capacidade de reconhecer mudanças no ambiente de operação ou nos objetivos e tomar ações corretivas. Um alto grau de adaptabilidade permite que uma organização seja mais ágil e resiliente, respondendo rapidamente a novos desafios e oportunidades e mantendo um alto nível de desempenho mesmo em face de mudanças e incertezas.

No contexto da calibração de motores, a adaptabilidade é uma fator muito importante quando analisamos o número de pessoas e sistemas envolvidos em cada um dos projetos. Toda e qualquer peça, sendo ela mecânica ou eletrônica tem imapcto direto no processo de calibração. Com a utilização da lista de tarefas, havia uma grande dificuldade em se saber quais eram os pontos de calibração que seriam afetados por cada uma dessas mudanças. Se por exemplo, houvesse uma alteração no coletor de admissão ou até mesmo no filtro de ar, identificar os testes, processamentos e pontos de calibração afetados seria uma tarefa bastante complexa. Os engenheiros deveriam buscar por inúmeros arquivos de documentação ou depender apenas de seu conhecimento prévio para, então, refazer todos os testes e adquirir os novos pontos de calibração.

Outro ponto bastante importate da adaptabilidade está na própria concepção do projeto. A reutilização de componentes e motores entre veículos de uma mesma montadora é uma prática comum e que ajuda na redução de custos e também pode diminuir a complexidade do processo de calibração. Se um motor será reutilizado, com apenas

algumas alterações, não se sabia quais eram os pontos que deveriam ser recalibrados e quais seriam os mesmos de projetos anteriores. A lista de tarefas não permitia, de maneira fácil e independente do engenheiro, eliminar pontos desnecessários.

1.2 Objetivos

O principal objetivo deste trabalho é implementar uma ferramenta baseada em grafos para melhorar o processo de calibração de motores, proporcionando maior eficiência e qualidade. Para mensurar a efetividade dessa implementação, para tal os seguintes objetivos específicos foram estabelecidos:

- Melhoria dos indicadores de desempenho
- Reduzir o tempo de ciclo
- Aumentar a eficiência do processo
- Possibilitar melhor visualização do uso de recursos

Ao atingir esses objetivos, esperamos demonstrar a efetividade da implementação do sistema baseado em grafos no processo de calibração de motores. O alcance dessas metas será medido e relatado no decorrer deste trabalho, a fim de fornecer uma avaliação clara e objetiva do impacto do sistema.

1.3 Organização do trabalho

O presente trabalho está estruturado para facilitar a compreensão do leitor sobre o tema proposto. No capítulo 2, a Contextualização proporciona um panorama mais amplo do cenário no qual a pesquisa está inserida.

O terceiro capítulo, denominado Fundamentação Teórica, é o coração do trabalho, abordando temas essenciais para o entendimento do estudo. Aqui, são discutidos em detalhe tópicos como Calibração de Motores, Teoria de Grafos — que é subdividida em várias subseções abrangendo definições, utilizações, métodos e aplicações específicas como Grafos em Gestão de Processos —, e ferramentas tecnológicas como Neo4j, JavaScript, e Node.js.

A Metodologia é explorada no capítulo 4, detalhando o processo de pesquisa, desde a identificação de especialistas até a implementação de visualizações gráficas em ferramentas web. Esta seção também destaca técnicas de gestão como a Análise de FMEA e indicadores-chave de desempenho.

O quinto capítulo, Análise de Resultados, oferece uma visão prática do trabalho, apresentando a ferramenta desenvolvida e discutindo sua aplicação e impacto por meio de técnicas de gestão.

Finalmente, no capítulo 6, as conclusões são delineadas, refletindo sobre os principais resultados e contribuições do trabalho. Também são apresentadas sugestões para Trabalhos Futuros, abrindo caminho para pesquisas subsequentes no campo.

O trabalho conclui com uma lista de Referências, fornecendo ao leitor todas as fontes bibliográficas utilizadas na elaboração deste estudo.

2 Fundamentação Teórica

2.1 Calibração de motores

A calibração de motores desempenha um papel fundamental na indústria automobilística, visando otimizar o desempenho, a eficiência energética e o controle de emissões de poluentes (HEYWOOD, 1988). Trata-se de um processo sistemático que envolve o ajuste dos parâmetros de controle do motor para garantir seu funcionamento ideal em diferentes condições de carga e velocidade.

Para iniciar o processo de calibração, é necessário definir os objetivos a serem alcançados, levando em consideração os requisitos específicos do motor e as regulamentações aplicáveis. Isso inclui estabelecer metas de desempenho, eficiência energética e limites de emissões (GUILLAUME; CHARLET, 2010). Em seguida, são coletados dados relevantes para a calibração, a partir da utilização de testes de bancada, simulações computacionais, testes em células de dinamômetros, diretamente nos veículos em pista, etc.

A análise dos dados coletados e modelagem do funcionamento do motor são etapas cruciais do processo de calibração. Utilizando modelos matemáticos, é possível compreender as complexas interações entre os diferentes componentes e parâmetros do motor (STONE, 1999). Esses modelos podem ser baseados em equações físicas, análise estatística ou técnicas de aprendizado de máquina. Essa análise permite identificar as melhores combinações de parâmetros que resultam em um desempenho otimizado do motor (JIN et al., 2018). Aqui estão alguns exemplos de curvas e equações que podem ser alteradas durante o processo de calibração:

1. Curva de Torque do Motor: É uma das principais curvas alteradas durante a calibração. Ela mostra a relação entre o torque do motor e a rotação do motor (RPM). Os calibradores podem alterar esta curva para atender às metas de desempenho e eficiência de combustível.

2. Mapa de Injeção de Combustível: Este mapa é uma matriz tridimensional que determina quanto combustível é injetado no motor em diferentes RPMs e cargas do motor. A calibração deste mapa é crucial para alcançar a relação ar-combustível ideal, que é fundamental para a eficiência de combustível e o controle de emissões. Para este controle, a equação geralmente envolve a massa de ar admitida no cilindro (determinada pelo sensor MAF ou MAP), a rotação do motor, a relação ar-combustível desejada e a densidade do combustível. Alterar os parâmetros desta equação pode ajustar a quantidade de combustível injetada.

$$m_f = \frac{m_a}{AFR} \cdot \frac{1}{\rho_f} \tag{2}$$

Onde:

- m_f é a massa de combustível injetada,
- m_a é a massa de ar admitida,
- AFR é a relação ar-combustível desejada,
- ρ_f é a densidade do combustível.

3. Mapa de Ignição: Este mapa, também uma matriz tridimensional, controla o momento da ignição (quando a vela de ignição dispara) em diferentes RPMs e cargas do motor. Alterar este mapa pode ter um impacto significativo no desempenho do motor e na eficiência do combustível. O tempo de ignição pode ser calculado com a seguinte equação:

$$t_i = \frac{\theta}{2 \cdot \pi \cdot N} \tag{3}$$

Onde:

- t_i é o tempo de ignição,
- θ é o ângulo de ignição (em radianos),
- N é a velocidade de rotação (em radianos por segundo).

4. Mapa de Controle de Válvulas: Em motores com controle de válvulas variável, este mapa determina o momento de abertura e fechamento das válvulas em diferentes RPMs e cargas do motor. A calibração deste mapa pode otimizar o enchimento do cilindro e a eficiência de combustível. O tempo de abertura das válvulas pode ser calculado com a seguinte equação:

$$t_v = \frac{\theta_v}{2 \cdot \pi \cdot N} \tag{4}$$

Onde:

- t_v é o tempo de abertura das válvulas,
- θ_v é o ângulo de abertura das válvulas (em radianos),
- N é a velocidade de rotação (em radianos por segundo).

Após a etapa de análise, é realizada a otimização dos parâmetros de calibração. Nessa fase, algoritmos de otimização são aplicados para encontrar a combinação ideal que atenda aos objetivos estabelecidos. Essa busca pode envolver a análise de um espaço multidimensional de parâmetros, levando em conta as restrições impostas pelos regulamentos e requisitos técnicos (DEB, 2001).

Com a otimização concluída, os ajustes de calibração são implementados no motor e são realizados testes de validação para verificar se os resultados desejados foram alcançados. Esses testes envolvem a avaliação do desempenho do motor em condições reais de operação, incluindo testes de rodagem e testes de emissões (COMMISSION, 2018).

É importante ressaltar que o processo de calibração é iterativo e passível de refinamentos. Com base nos resultados dos testes de validação, é possível realizar ajustes adicionais para melhorar ainda mais o desempenho, a eficiência e a redução de emissões. Essa iteração continua até que os resultados desejados sejam alcançados e atenda aos requisitos estabelecidos (DEB; DEB, 2011).

Este processo também está sujeito ao uso de ferramentas avançadas de medição e análise, a disponibilidade de equipamentos de teste e validação adequados, além do cumprimento das regulamentações e normas aplicáveis (PUNDIR, 2016).

A implementação de ferramentas baseadas em tecnologias como JavaScript e bancos de dados como o Neo4j oferece a oportunidade de integrar sistemas de calibração mais eficientes e sustentáveis. Essas ferramentas permitem a manipulação de grandes volumes de dados, possibilitando uma análise mais abrangente e uma calibração mais precisa dos motores.

Além disso, a utilização de técnicas de inteligência artificial, como redes neurais profundas e algoritmos de aprendizado de máquina, mostram-se promissores para melhorar ainda mais o processo de calibração de motores. Essas técnicas têm a capacidade de analisar padrões complexos nos dados coletados, identificar relações não lineares e otimizar os ajustes de calibração para alcançar melhores resultados (PAL et al., 2023).

Em suma, este é um processo complexo e essencial na indústria automobilística. Através da análise de dados, modelagem matemática, otimização e adoção de tecnologias avançadas, é possível aprimorar o desempenho, a eficiência energética e o controle de emissões dos motores, atendendo às demandas do mercado e contribuindo para um futuro mais sustentável.

2.2 Teoria de Grafos

A teoria de grafos é um campo matemático que estuda as propriedades e as relações entre os grafos, que são estruturas compostas por um conjunto de vértices (ou nós) e um conjunto de arestas (ou conexões) que ligam esses vértices. Essa área de estudo tem diversas aplicações em ciência da computação, logística, redes sociais e outros campos, permitindo a modelagem e a solução de problemas complexos (WEST, 2001).

2.2.1 Definição de Grafos

Formalmente, um grafo é um par ordenado:

$$G = (V, E) \tag{5}$$

Em que:

- V representa o conjunto de vértices,
- E representa o conjunto de arestas.

Um grafo pode ser direcionado, em que as arestas possuem uma orientação específica, ou não direcionado, em que as arestas não têm uma direção específica. Além disso, as arestas podem ser ponderadas, atribuindo um valor numérico a cada aresta (DIESTEL, 2017).

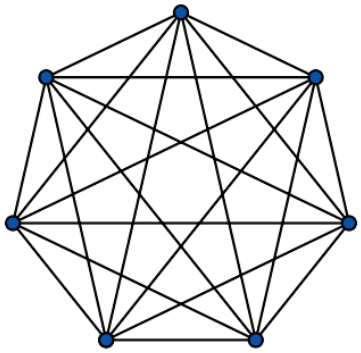

Figura 3 – Grafo Completo

Um exemplo simples de grafo é o Grafo Completo, em que todos os vértices estão conectados entre si por uma aresta. Esse grafo é representado por K_n, onde n é o número de vértices 3. Já o Grafo Vazio, representado por E_n, é aquele em que não há arestas, apenas vértices isolados.

No contexto deste trabalho, o grafo em questão possui características específicas que são fundamentais para compreender a aplicação dos algoritmos e metodologias abordadas. As principais características do grafo são:

- Dígrafo (Grafo Direcionado): O grafo utilizado é um dígrafo, o que significa que todas as arestas possuem uma direção associada. Em outras palavras, cada aresta move-se de um vértice de origem para um vértice de destino, e essa direção é fundamental para a representação e análise do problema. Dígrafos são particularmente úteis em contextos onde as relações possuem uma natureza direcional, como precedência de tarefas em gerenciamento de projetos.

- Conexo: Um grafo conexo é aquele em que existe um caminho entre qualquer par de vértices. No contexto deste trabalho, a conexidade garante que todas as partes ou componentes do sistema estão interligadas de alguma forma, direta ou indiretamente. Isso é crucial para garantir que o grafo representa um sistema coeso, onde todas as entidades (vértices) têm relevância e estão interconectadas no contexto geral do problema.

A combinação dessas propriedades - direcionalidade, ausência de ponderação e conexidade - molda a forma como o problema é representado e analisado neste trabalho. Cada característica traz implicações importantes para a metodologia de análise e para a aplicação dos algoritmos discutidos.

2.2.2 Utilização de Grafos

Os grafos são amplamente utilizados para modelar e resolver uma variedade de problemas complexos. Eles fornecem uma representação visual e matemática eficiente para a análise de relacionamentos e interações entre elementos. Dentre as aplicações, destacam-se:

- **Redes Sociais e Amizades:** Grafos são usados para modelar conexões entre pessoas em redes sociais, permitindo a análise de amizades, influências sociais e propagação de informações (NEWMAN, 2018).
- **Transporte e Logística:** Grafos podem representar rotas, conexões e fluxos de transporte em uma rede, auxiliando na otimização de rotas, distribuição de recursos e planejamento logístico.
- **Computação e Algoritmos:** Grafos são fundamentais para algoritmos de busca, como o algoritmo de Dijkstra e o algoritmo de busca em largura (BFS), amplamente utilizados em problemas de otimização e processamento de dados (CORMEN et al., 2009) (DIJKSTRA, 1959).
- **Sistemas de Recomendação:** Grafos podem ser usados para construir sistemas de recomendação personalizados, identificando conexões entre itens e usuários para sugerir recomendações relevantes.
- **Biologia e Genômica:** Grafos são usados na análise de redes de interação de proteínas, redes de regulação genética e outras relações biológicas complexas.

2.2.3 Métodos em Teoria de Grafos

Existem vários métodos e algoritmos que permitem a análise e a resolução de problemas específicos. Um desses métodos é o Método do Caminho Crítico (CPM -

Critical Path Method), é um método utilizado para gerenciar e otimizar o cronograma de um projeto. Ele se baseia na análise dos caminhos críticos do projeto, que são as sequências de atividades que determinam a duração total do projeto (KERZNER, 2017b). O CPM é amplamente aplicado em áreas como engenharia, construção civil, gerenciamento de projetos e logística.

O objetivo do CPM é identificar o caminho crítico, ou seja, a sequência de atividades que não podem atrasar sem afetar o tempo total do projeto. Essas atividades têm uma folga total de zero, o que significa que qualquer atraso nelas resultará em um atraso no projeto como um todo. Por outro lado, atividades que não estão no caminho crítico possuem alguma folga, ou seja, podem ser atrasadas sem impactar o prazo final do projeto.

2.2.4 Algoritmos de grafos

- Programação Linear (PL): A Programação Linear (PL) é uma técnica que busca otimizar (maximizar ou minimizar) uma função linear sujeita a restrições também lineares. A ideia é encontrar o ponto no espaço que atende a todas as restrições e otimiza a função objetivo. Em contextos de gestão de projetos, a PL pode ser utilizada para otimizar a alocação de recursos em diversas atividades, mantendo o custo total mínimo ou o lucro máximo.

- Programação Dinâmica: Diferentemente de métodos que reavaliam o mesmo problema várias vezes, a Programação Dinâmica procura solucionar cada subproblema apenas uma vez, guardando sua solução em uma tabela (geralmente implementada como uma matriz ou algum tipo de mapa), evitando o trabalho de recomputar a solução toda vez que o subproblema é encontrado. Esta técnica é muito útil em problemas de sequenciamento e otimização de rotas, comuns em gestão de projetos.

- *Program Evaluation Review Technique* PERT: O PERT é semelhante ao CPM no sentido de que ambos se concentram no sequenciamento e otimização de tarefas em um projeto. No entanto, o PERT leva em consideração a incerteza e variabilidade, utilizando três estimativas para a duração de cada atividade: a mais otimista, a mais provável e a mais pessimista. Ao contrário do CPM, que utiliza uma única estimativa determinística, o PERT fornece uma abordagem probabilística ao gerenciamento de projetos.

- Programação de Inteiros: Enquanto a Programação Linear se concentra em soluções contínuas, a Programação de Inteiros destina-se a problemas de otimização onde as soluções devem ser valores discretos. Por exemplo, em um contexto de

gerenciamento de projetos, pode ser usado para determinar o número mínimo de trabalhadores necessários para completar um conjunto de tarefas no prazo.

- Algoritmo de Dijkstra: Originalmente proposto por Edsger W. Dijkstra em 1956, este algoritmo resolve o problema do caminho mais curto em um grafo. Em termos de gerenciamento de projetos, pode ser usado para encontrar a rota mais eficiente em termos de tempo ou custo entre dois pontos (ou tarefas) em um projeto representado como um grafo. Ao otimizar essas rotas, é possível economizar recursos e tempo(DIJKSTRA, 1959).

2.2.5 O método CPM

O CPM utiliza algumas fórmulas matemáticas para calcular as datas de início mais cedo (ES, do inglês Early Start), as datas de término mais cedo (EF, do inglês Early Finish), as datas de início mais tarde (LS, do inglês Late Start) e as datas de término mais tarde (LF, do inglês Late Finish) de cada atividade. Essas informações são usadas para determinar o caminho crítico e a duração total do projeto.

As fórmulas principais do CPM são:

- **Cálculo do ES e EF (passagem para frente)**
  - O ES de uma atividade é igual ao EF da atividade anterior (ou atividades, se houver mais de uma atividade anterior).
  - Se houver mais de uma atividade anterior, o ES é igual ao maior EF entre as atividades anteriores.
  - O EF é calculado somando a duração da atividade ao ES.
- **Cálculo do LS e LF (passagem para trás)**
  - O LF de uma atividade é igual ao LS da atividade seguinte (ou atividades, se houver mais de uma atividade seguinte).
  - Se houver mais de uma atividade seguinte, o LF é igual ao menor LS entre as atividades seguintes.
  - O LS é calculado subtraindo a duração da atividade do LF.

Note que a primeira atividade do projeto tem seu ES igual a 0, e a última atividade tem seu LF igual ao EF da última atividade. E na passagem para trás, se uma atividade não tiver atividades dependentes, seu LF é igual ao LF do projeto (ou seja, ao EF da última atividade).

Por último, a folga total de uma atividade pode ser calculada como LF - EF ou LS - ES, que te dão o mesmo resultado. A folga total indica quanto tempo uma atividade pode ser atrasada sem atrasar o projeto.

A fim de facilitar a compreensão, determina-se o seguinte exemplo demonstrado em 4.

	Predecessores	Duração	ES	EF	LS	LF	1	2	3	4	5	6
Tarefa 1		1	0	1	0	1						
Tarefa 2	1	1	1	2	3	4						
Tarefa 3	1	3	1	4	1	4						
Tarefa 4	2, 3	1	4	5	4	5						
Tarefa 5	4	1	5	6	5	6						
Tarefa 6	2, 4	1	5	6	5	6						

Figura 4 – Gráfico de gantt, e detalhamento do caminho crítico de projeto exemplo

Vamos começar com a Tarefa 1. Como é a primeira tarefa, o ES é 0. Considerando que a duração é 1, logo EF = 1.

Agora, vamos considerar a Tarefa 2, cujo predecessor é a Tarefa 1. Assim, o ES para a Tarefa 2 é o EF da Tarefa 1, que é 1. A duração da Tarefa 2 é 1, logo EF = 2.

Analogamente para a Tarefa 3, que também tem a Tarefa 1 como predecessor, ES = 1 (EF da Tarefa 1), e como sua duração é 3, logo EF = 4.

A Tarefa 4 tem como predecessores a Tarefa 2 e a Tarefa 3. Neste caso, o ES é o maior EF entre os predecessores, que é 4 (EF da Tarefa 3). Com duração de 1, logo EF = 5.

A Tarefa 5 tem como predecessor a Tarefa 4. Portanto, o ES é 5 (EF da Tarefa 4) e com a duração de 1, logo EF = 6.

A Tarefa 6 tem como predecessores a Tarefa 2 e a Tarefa 4. Neste caso, novamente o ES é o maior EF entre os predecessores, que é 5 (EF da Tarefa 4). Com duração de 1, logo EF = 6.

Agora, vamos ao cálculo do LS e LF, que é feito de maneira reversa, começando pela última tarefa.

Como a Tarefa 6 é a última tarefa, o LF é igual ao EF, que é 6. Com a duração de 1, logo LS = 5.

A Tarefa 5 também não possui dependentes. Portanto, o LF é igual ao EF, que é 6. Com a duração de 1, logo LS = 5.

A Tarefa 4 tem a Tarefa 6 como dependente. Portanto, o LF é igual ao LS da Tarefa 6, que é 5. Com a duração de 1, logo LS = 4.

As Tarefas 2 e 3 têm como dependentes as Tarefas 4 e 6. O LF para ambas as tarefas será o menor LS entre os dependentes, que é 4. Com as durações de 1 e 3

respectivamente, LS para a Tarefa 2 é LS = 3, e para a Tarefa 3 LS = 1.

Finalmente, a Tarefa 1 tem como dependentes as Tarefas 2 e 3. O LF para a Tarefa 1 será o menor LS entre os dependentes, que é 1. Com duração de 1, logo LS = 0.

Pro fim, a única tarefa que não compõe o caminho crítico desse projeto é a tarefa 2, que possui folga de duração igual a 2.

2.2.6 Requisitos do Método CPM

Para aplicar o método CPM, é necessário seguir alguns requisitos:

- **Identificação das atividades:** É preciso identificar todas as atividades necessárias para a conclusão do projeto e definir suas durações.
- **Relações de dependência:** Deve-se estabelecer as relações de dependência entre as atividades, ou seja, determinar quais atividades precisam ser concluídas antes que outras possam começar.
- **Construção do diagrama de rede:** O próximo passo é construir o diagrama de rede, também conhecido como Diagrama de Precedência e Sucessão (PDM, do inglês Precedence Diagramming Method), que representa as atividades como nós e as relações de dependência como arestas direcionadas (MORTON; PENTICO, 2014).
- **Cálculo das datas de início e término:** Com o diagrama de rede construído, é possível calcular as datas de início mais cedo (ES), término mais cedo (EF), início mais tarde (LS) e término mais tarde (LF) de cada atividade, utilizando as fórmulas do CPM mencionadas anteriormente.
- **Determinação do caminho crítico:** Com as datas de início e término calculadas, é possível identificar o caminho crítico do projeto. O caminho crítico é composto pelas atividades cuja folga total é igual a zero, ou seja, qualquer atraso nessas atividades resultará em um atraso no projeto como um todo.
- **Otimização do cronograma:** Uma vez identificado o caminho crítico, é possível focar nos recursos e atividades críticas para garantir que elas sejam concluídas dentro do prazo. Isso envolve a alocação adequada de recursos, o gerenciamento de riscos e a otimização do fluxo de trabalho.
- **Atualização do cronograma:** Durante a execução do projeto, é importante monitorar o progresso das atividades e fazer ajustes no cronograma, caso necessário. Eventuais atrasos ou mudanças nas dependências entre as atividades podem exigir uma atualização do caminho crítico e das datas de término do projeto.

O método CPM é amplamente utilizado em projetos de grande escala, onde a gestão eficiente do tempo é essencial. Ele fornece uma visão clara das atividades críticas e suas interdependências, permitindo que os gerentes de projeto tomem decisões informadas para garantir o cumprimento dos prazos.

É importante destacar que o CPM não considera recursos limitados e assume que as atividades serão executadas com a maior eficiência possível. Portanto, é necessário realizar ajustes adicionais para levar em conta restrições de recursos e outras variáveis do projeto (KERZNER, 2017b).

Além do CPM, existem outros métodos relacionados, como o Método do Caminho Mais Curto (Shortest Path Method) e o Método de Fluxo Máximo (Maximum Flow Method), que têm aplicações específicas em diferentes contextos (NEWMAN, 2018).

2.2.7 Grafos em Gestão de Processos e Controle de Qualidade

Grafos tem sido aplicados em vários aspectos da gestão de processos e controle de qualidade. Um exemplo é a aplicação de grafos na análise de redes de produção para identificar gargalos e otimizar o fluxo de materiais e informações (KIM; PARK, 2007). Outro exemplo é o uso de grafos na modelagem e análise de processos de negócios, onde os vértices representam atividades e as arestas representam relacionamentos entre essas atividades (PETEGHEM; VANHOUCKE, 2008).

No setor automotivo, a teoria tem sido empregada no planejamento e controle de processos de produção para melhorar a eficiência e a qualidade dos produtos. Por exemplo, DAI et al. (DAI et al., 2018) apresentam um estudo de caso em que aplicam grafos para analisar e otimizar a rede de montagem de veículos em uma fábrica de automóveis. Neste estudo, os autores identificam gargalos e propõem soluções para melhorar o desempenho e a qualidade do processo de montagem.

Um exemplo notável no setor automotivo é o estudo de caso da Volvo, em que STJERNSTRÖM; HOLMSTRÖM (STJERNSTRÖM; HOLMSTRÖM, 2003) descrevem o design e a implementação de um sistema de controle de fluxo de materiais flexível baseado em teoria de grafos e heurísticas. Neste estudo, os autores aplicam a os métodos para gerenciar e controlar o processo de fabricação de veículos, melhorando a eficiência e a qualidade da produção.

2.2.8 Grafos em Calibração de Motores

A aplicação de Grafos de projeto no contexto da calibração de motores é uma área de pesquisa emergente. Embora a literatura específica sobre esse assunto seja limitada, pode-se argumentar que o uso de grafos para representar e analisar a estrutura e as dependências entre as variáveis de calibração pode ser benéfico (KLIJN et al., 2021). A

modelagem de grafos pode ajudar a identificar e priorizar variáveis-chave no processo de calibração, facilitando a otimização do processo.

Além disso, grafos têm sido aplicados com sucesso em outras áreas relacionadas à engenharia, como otimização de rotas, gerenciamento de projetos e logística (CINPOERU et al., 2019) (WORKFLOW. . . , 2012). Essas aplicações mostram o potencial dos grafos para modelar problemas complexos e encontrar soluções eficientes.

No contexto da calibração de motores, uma abordagem baseada em grafos pode ser útil para identificar gargalos e otimizar o fluxo de informações entre as variáveis de calibração. Isso pode resultar em melhorias no processo de calibração e, consequentemente, na qualidade e no desempenho dos motores.

2.3 Neo4j

O Neo4j é um banco de dados de grafos altamente escalável e de alto desempenho, desenvolvido para armazenar e gerenciar dados representados em forma de grafo (NEO4J. . . ,). Foi lançado pela primeira vez em 2007 por Emil Eifrem, Johan Svensson e Peter Neubauer, e desde então tem se tornado cada vez mais popular entre desenvolvedores e cientistas de dados que trabalham com grafos e análises de redes (ROBINSON et al., 2013). A principal vantagem do Neo4j em comparação com outros sistemas de gerenciamento de banco de dados é sua capacidade de armazenar e manipular dados de grafo de forma nativa, o que significa que o armazenamento e as operações realizadas nos dados são otimizados para a estrutura de grafo. Isso resulta em um melhor desempenho e escalabilidade, especialmente em consultas complexas e análises de redes (NEO4J. . . ,). Uma característica distintiva do Neo4j é sua linguagem de consulta, chamada Cypher. Cypher é uma linguagem de consulta declarativa e específica de domínio, projetada especificamente para trabalhar com grafos no Neo4j (ROBINSON et al., 2013). A sintaxe do Cypher é expressiva e fácil de entender, permitindo aos usuários escrever consultas complexas de uma maneira intuitiva e legível. Além disso, o Cypher oferece suporte a padrões de correspondência de grafos, permitindo que os usuários especifiquem padrões de conexões entre nós e relacionamentos e, em seguida, busquem esses padrões no grafo (CYPHER. . . ,).

2.3.1 Sintaxe e exemplos de consultas Cypher

A sintaxe do Cypher é projetada para ser intuitiva e fácil de entender, usando padrões de ASCII-art para representar grafos e relacionamentos (CYPHER. . . ,). A seguir, são apresentadas algumas das principais palavras-chave e conceitos da linguagem Cypher, juntamente com exemplos de consultas:

- **MATCH**: A cláusula MATCH é usada para especificar um padrão de grafo que queremos encontrar no banco de dados. Os nós são representados por parênteses e os relacionamentos por colchetes. Por exemplo, para encontrar todos os nós que têm um relacionamento "AMIGO"com o nó "Lucas":

```
MATCH (a)-[:AMIGO]->(b {name: "Lucas"})
RETURN a
```

- **CREATE**: A cláusula CREATE é usada para criar novos nós e relacionamentos no banco de dados. Por exemplo, para criar um novo nó "Carla"e relacioná-lo com "Lucas"por meio de um relacionamento "AMIGO":

```
CREATE (a {name: "Carla"})-[:AMIGO]->(b {name: "Lucas"})
```

- **WHERE**: A cláusula WHERE é usada para adicionar condições às consultas. Por exemplo, para encontrar todos os amigos de "Lucas"com idade maior que 25 anos:

```
MATCH (a)-[:AMIGO]->(b {name: "Lucas"})
WHERE a.age > 25
RETURN a
```

- **RETURN**: A cláusula RETURN é usada para especificar quais informações devem ser retornadas como resultado da consulta. Pode-se retornar propriedades específicas ou todo o nó. Por exemplo, para retornar apenas os nomes dos amigos de "Lucas":

```
MATCH (a)-[:AMIGO]->(b {name: "Lucas"})
RETURN a.name
```

- **DELETE e DETACH DELETE**: As cláusulas DELETE e DETACH DELETE são usadas para excluir nós e relacionamentos do banco de dados. DELETE remove apenas o relacionamento, enquanto DETACH DELETE remove o nó e todos os seus relacionamentos. Por exemplo, para remover um relacionamento "AMIGO"entre "Carla"e "Lucas":

```
MATCH (a {name: "Carla"})-[r:AMIGO]->(b {name: "Lucas"})
DELETE r
```

  Para remover o nó "Carla"e todos os seus relacionamentos:

```
MATCH (a {name: "Carla"})
DETACH DELETE a
```

Esses exemplos ilustram algumas das funcionalidades básicas da linguagem Cypher. Há muito mais a aprender sobre Cypher, incluindo agregação, ordenação, funções e outras cláusulas avançadas, como:

- **WITH**: A cláusula WITH permite encadear consultas e passar os resultados de uma consulta para outra. Por exemplo, para encontrar todos os amigos de "Lucas"e contar quantos amigos eles têm:

```
MATCH (a)-[:AMIGO]->(b {name: "Lucas"})
WITH a, count(a) as num_amigos
RETURN a.name, num_amigos
```

- **ORDER BY**: A cláusula ORDER BY é usada para ordenar os resultados retornados por uma consulta. Por exemplo, para retornar todos os amigos de "Lucas"em ordem alfabética:

```
MATCH (a)-[:AMIGO]->(b {name: "Lucas"})
RETURN a.name
ORDER BY a.name
```

- **SKIP e LIMIT**: As cláusulas SKIP e LIMIT são usadas para controlar a quantidade de resultados retornados por uma consulta. Por exemplo, para retornar os três primeiros amigos de "Lucas"em ordem alfabética:

```
MATCH (a)-[:AMIGO]->(b {name: "Lucas"})
RETURN a.name
ORDER BY a.name
SKIP 0
LIMIT 3
```

Essas cláusulas adicionais demonstram a flexibilidade e o poder da linguagem Cypher, permitindo a criação de consultas complexas e personalizadas para explorar e manipular dados em um banco de dados Neo4j. Além disso, a comunidade Neo4j oferece uma ampla gama de recursos e tutoriais para ajudar a aprender e dominar a linguagem Cypher (NEO4J, 2023).

2.4 JavaScript e Node.js

JavaScript é uma linguagem de programação versátil e amplamente utilizada, principalmente no desenvolvimento de aplicações web. Originalmente concebida para rodar no lado do cliente (navegadores web), a linguagem evoluiu e expandiu-se para o desenvolvimento no lado do servidor, graças ao surgimento do Node.js (FLANAGAN, 2020).

2.4.1 JavaScript

JavaScript é uma linguagem de programação de alto nível, dinâmica, interpretada e orientada a objetos, com suporte a múltiplos paradigmas, como programação imperativa e funcional. Sua popularidade e ampla adoção se devem, em grande parte, à sua capacidade de permitir interações dinâmicas e ricas entre o usuário e o navegador web, tornando a experiência na web mais interessante e envolvente (FLANAGAN, 2020).

Embora a linguagem seja frequentemente associada ao desenvolvimento de aplicações web, suas capacidades não se limitam a esse domínio. JavaScript também tem sido aplicada em outras áreas, como desenvolvimento de aplicativos móveis e desktop, desenvolvimento de jogos, automação e Internet das Coisas (IoT) (KOSMACZEWSKI, 2012).

2.4.2 Node.js

Node.js é uma plataforma de execução de código JavaScript construída sobre o motor JavaScript V8 do Google Chrome. Foi criado por Ryan Dahl em 2009 e desde então, ganhou popularidade por permitir que os desenvolvedores executem JavaScript no lado do servidor (TILKOV; VINOSKI, 2010). Esta plataforma tem como principal característica a capacidade de realizar operações assíncronas e não bloqueantes, o que a torna especialmente adequada para aplicações escaláveis e de alta performance.

Node.js utiliza um modelo de I/O (entrada e saída) baseado em eventos, permitindo que as aplicações lidem com múltiplas conexões simultâneas sem a necessidade de criar uma thread para cada conexão. Isso resulta em um melhor aproveitamento dos recursos do sistema e maior capacidade de atender a um maior número de usuários simultâneos.

A plataforma Node.js inclui um gerenciador de pacotes chamado npm (Node Package Manager), que facilita o gerenciamento de bibliotecas e dependências de projetos. O ecossistema npm é vasto e possui uma infinidade de pacotes disponíveis, fornecendo soluções para praticamente todas as necessidades de desenvolvimento

(CANTELON et al., 2013).

Ao combinar JavaScript com Node.js, os desenvolvedores podem criar aplicações completas, desde o lado do cliente até o lado do servidor, utilizando uma única linguagem de programação. Isso simplifica o processo de desenvolvimento e facilita a manutenção e a integração de novas funcionalidades.

2.4.3 ForceGraph2D

A biblioteca ForceGraph2D é uma ferramenta JavaScript para visualização de grafos interativos em duas dimensões. Baseada na biblioteca D3.js(BOSTOCK, 2021), ela utiliza simulações de forças para posicionar nós e arestas de um grafo, fornecendo uma representação gráfica, como em 5 intuitiva e dinâmica(SILVA, 2021).

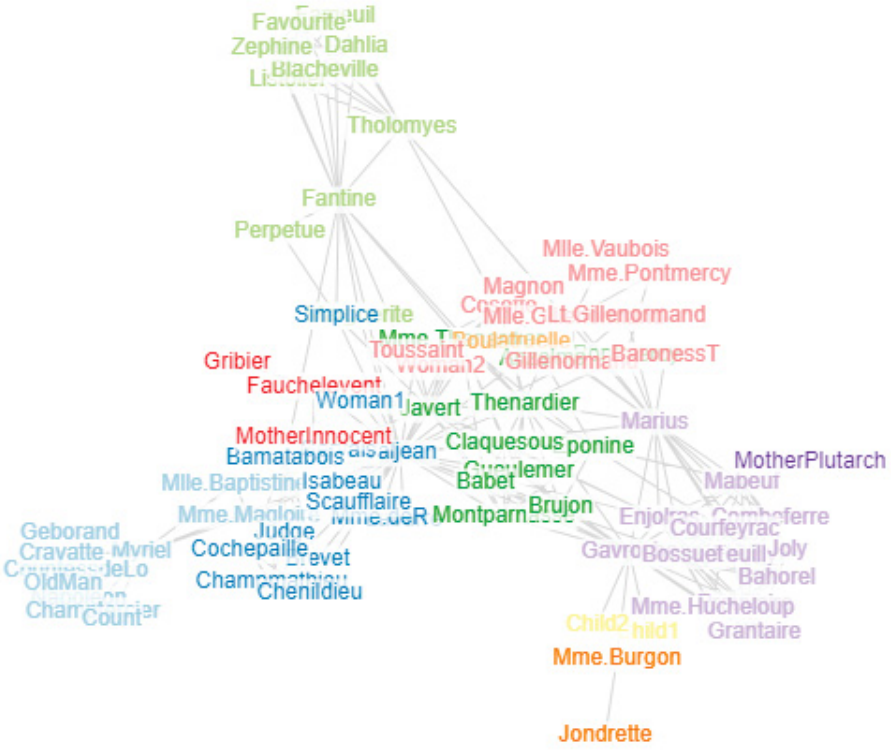

Figura 5 – Exemplo de grafo plotado com a ajuda da biblioteca Force Graph 2D

A principal vantagem do ForceGraph2D é a sua capacidade de lidar com grafos grandes e complexos, possibilitando interações do usuário com os nós e arestas em tempo real. Além disso, a biblioteca oferece várias opções de personalização, como cores, tamanhos e estilos dos elementos do grafo.

O ForceGraph2D pode ser facilmente integrado a projetos JavaScript e é compatível com bibliotecas populares, como React e Vue.js. Algumas aplicações típicas da

ForceGraph2D incluem análise de redes sociais, visualização de sistemas biológicos e representação de estruturas de dados complexas.

2.4.4 Echarts

Echarts é uma biblioteca JavaScript de visualização de dados de código aberto desenvolvida pela Baidu(BAIDU, 2021a). Ela oferece uma ampla variedade de gráficos, incluindo gráficos de barras, linhas, áreas, dispersão, radar, torta e muitos outros. Além disso, a biblioteca Echarts também suporta visualizações de grafos e redes(BAIDU, 2021b).

A principal característica do Echarts é sua flexibilidade e facilidade de uso. Os gráficos podem ser facilmente configurados por meio de opções de estilo e dados, permitindo aos desenvolvedores criar visualizações personalizadas sem a necessidade de escrever código complexo. Além disso, o Echarts é responsivo e adaptável a diferentes tamanhos de tela e dispositivos, garantindo uma experiência de usuário consistente em diferentes plataformas.

Echarts pode ser integrado a várias bibliotecas e frameworks JavaScript, como React, Angular e Vue.js, facilitando a inclusão de visualizações de dados interativas em aplicações web. Essa biblioteca é amplamente utilizada em diversos setores, como finanças, ciência de dados, análise de redes e monitoramento de desempenho.

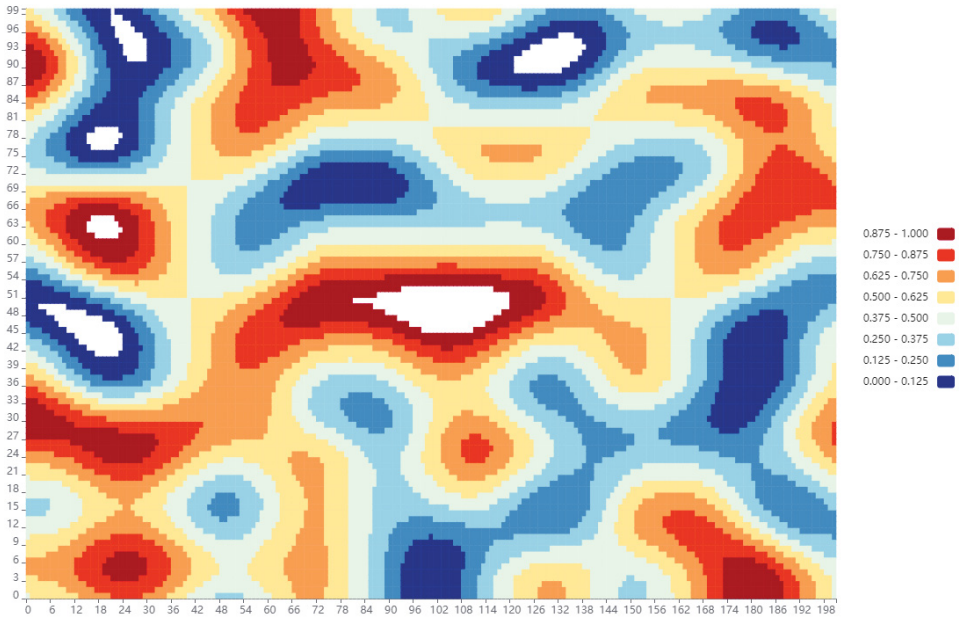

Figura 6 – Exemplo de gráfico plotado com a ajuda da biblioteca Echarts

3 Desenvolvimento

A metodologia acadêmica adotada para esta pesquisa foi estruturada em seis etapas principais

1. Revisão bibliográfica
2. Identificação de especialistas
3. Coleta de dados
4. Modelagem do banco
5. Implementação da Visualização do Grafo em uma Ferramenta Web
6. Interface de buscas
7. Análise de caminho crítico
8. Análise dos resultados obtidos

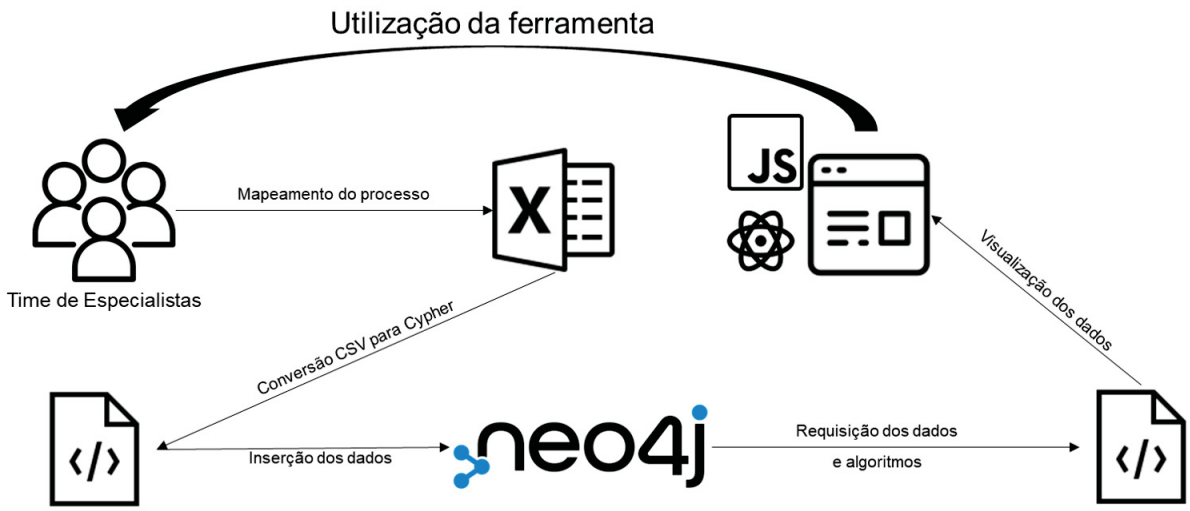

Figura 7 – Arquitetura da ferramenta

3.1 Identificação de especialistas

Um dos primeiros e mais importantes passos na implementação deste projeto foi a identificação dos especialistas adequados dentro da organização. Dada a complexidade do processo de calibração de motores no setor automotivo, tornou-se evidente

que um entendimento profundo e experiência prática seriam necessários para criar uma representação precisa e útil desse processo na forma de grafos.

A identificação dos especialistas foi realizada através de uma série de etapas. Primeiramente, foi realizada uma análise organizacional para identificar os departamentos e funções que estão diretamente envolvidos no processo de calibração de motores. Esta análise levou em consideração tanto as funções formais e hierárquicas quanto as relações e interações informais que ocorrem no dia a dia da organização.

Em seguida, foram conduzidas entrevistas com gestores e líderes de equipe para identificar os indivíduos que possuem um conhecimento profundo e experiência prática no processo de calibração de motores. Foi dada especial atenção aos funcionários que demonstraram uma compreensão clara do processo como um todo, e não apenas de suas partes individuais.

Finalmente, os especialistas identificados foram convidados a participar do projeto, contribuindo com seus conhecimentos e experiência para a criação do grafo. Este processo não foi apenas uma questão de coleta de informações, mas também um exercício de colaboração e engajamento. Os especialistas foram incentivados a discutir e debater sobre o processo, identificar pontos de melhoria, e sugerir possíveis soluções.

A identificação e envolvimento dos especialistas não só aumentou a precisão e utilidade do grafo, mas também contribuiu para um maior engajamento e aceitação do projeto dentro da organização. Ao envolver os especialistas desde o início, o projeto ganhou credibilidade e relevância, aumentando as chances de sua implementação bem-sucedida e adoção a longo prazo.

3.2 Coleta de dados

A coleta de dados é uma etapa fundamental na implementação deste projeto. Estes dados são a base para a criação do grafo que representa o processo de calibração de motores. Para facilitar a coleta de dados e garantir a consistência e precisão das informações, foi proposto um formato de planilha Excel para ser preenchido pelos especialistas.

A planilha foi dividida em várias seções para capturar as diferentes dimensões do processo. As principais seções incluem:

Nós: Esta seção é destinada a capturar os diferentes elementos ou "nós"do processo. Cada linha nesta seção representa um nó, com colunas para o nome do nó, descrição, tipo de nó (por exemplo, tarefa, decisão, etc.), e outros atributos relevantes.

Relações: Esta seção é destinada a capturar as relações entre os nós. Cada linha nesta seção representa uma relação, com colunas para o nó de origem, nó de destino,

tipo de relação (por exemplo, precede, segue, etc.), e outros atributos relevantes.

Atributos: Esta seção é destinada a capturar os atributos adicionais dos nós e relações que não são capturados nas seções anteriores. Isso pode incluir, por exemplo, a duração estimada de uma tarefa, o responsável por uma tarefa, o custo associado a uma tarefa, etc.

3.3 Modelagem do banco

Os dados coletados na planilha Excel serão então usados como entrada para um algoritmo JavaScript. Este algoritmo será responsável por converter os dados em uma série de queries Cypher que serão usadas para criar os nós e relações no banco de dados Neo4j. Por exemplo, uma linha na seção "Nós"da planilha pode ser convertida em uma query Cypher como a seguinte:

```
CREATE (:Task {
    name: 'Calibration',
    description: 'Calibration of the engine',
    duration: '3 days'})
```

Da mesma forma, uma linha na seção "Relações"da planilha pode ser convertida em uma query Cypher como a seguinte:

```
MATCH (a:Task {name: 'Calibration'}),
    (b:Task {name: 'Testing'})
CREATE (a)-[:PRECEDES]->(b)
```

Desta forma, a coleta de dados e a criação do grafo são integradas em um único processo, facilitando a implementação e garantindo a precisão e consistência dos dados.

3.4 Implementação da Visualização do Grafo em uma Ferramenta Web

A visualização do grafo é uma etapa essencial para entender e analisar os dados adquiridos. Para a implementação desta visualização, foi utilizada uma combinação de JavaScript (JS), React, ForceGraph2D e Dagre.

A ferramenta de vizualição de processos utilizando grafos foi idealizada para complementar um website já existente dentro da empresa, portanto, a escolha pela

linguagem JavaScript e pelo Framework React.js seguiram o que já estava sendo implementado na ferramenta. Porém é possível vizualizarmos uma matriz de escolha pensando em trabalhos futuros.

	JS/ React	Python/ Flask	Java/ Spring	Ruby on Rails
Facilidade de aprendizado	Alta	Alta	Média	Baixa
Suporte à biblioteca de visualização	Alta	Média	Baixa	Média
Integração com Neo4j	Alta	Alta	Alta	Média

Tabela 1 – Matriz de escolha de linguagem/framework para visualização e integração com Neo4j.

A seguir, mostraremos uma série de demonstrações de como as bibliotecas foram utilizadas e o que cada porção de código representa. Assim, fornecendo um exemplo de como o JavaScript e as biblioteccas mencionadas podem ser usadas juntas para criar uma visualização de grafo.

3.5 Dagre

No trecho de código a baixo, podemos observar como a biblioteca Dagre define os nós e links dos dados retornados pelo banco

```
import React from 'react';
import { ForceGraph2D } from 'react-force-graph';
import dagre from 'dagre';

// Criando um grafo direcional
let g = new dagre.graphlib.Graph();
g.setGraph({});
g.setDefaultEdgeLabel(() => ({}));

// Adicionando nós ao grafico
data.nodes.forEach(node => {
  g.setNode(node.id, { label: node.label });
});

// Adicionando relações ao gráfico
data.edges.forEach(edge => {
  g.setEdge(edge.source, edge.target);
});
```

```
// Reorganizando os elementos na tela
dagre.layout(g);
```

Este código demonstra como os dados são carregados, um novo grafo é criado, nós e arestas são adicionados ao grafo, o algoritmo de layout Dagre é executado, e finalmente, o componente ForceGraph2D é criado. O resultado é uma visualização interativa do grafo que pode ser facilmente explorada e analisada.

3.6 Análise de Impacto e Melhoria de Processos

Após a implementação e o estabelecimento da ferramenta de visualização de grafos, a próxima etapa crítica é avaliar o impacto e as melhorias nos processos. Esta avaliação é vital para entender o valor e a eficácia do sistema proposto.

3.6.1 Análise de FMEA

A Análise de Modo e Efeitos de Falha (FMEA, do inglês Failure Mode and Effects Analysis) é uma técnica eficaz que pode ser utilizada para avaliar o impacto da ferramenta e dos grafos. Essencialmente, a FMEA é um método estruturado para descobrir possíveis falhas no projeto e seus possíveis efeitos. Esta análise pode ajudar a identificar, priorizar e mitigar riscos (STAMATIS, 2003).

Os passos envolvidos na execução de uma FMEA incluem:

1. Identificação dos modos de falha
2. Determinação dos efeitos de cada falha
3. Avaliação da severidade, ocorrência e detecção de cada falha
4. Cálculo do Índice de Prioridade de Risco (RPN)
5. Desenvolvimento e implementação de ações para reduzir o RPN

A FMEA é uma ferramenta versátil que pode ser utilizada não apenas para identificar e mitigar possíveis falhas, mas também para comparar diferentes cenários. Neste caso, podemos usar a FMEA para comparar a situação antes e depois da implementação da ferramenta.

O primeiro passo para realizar essa comparação é executar uma FMEA para a situação "antes". Isso envolve a identificação dos modos de falha, a determinação dos

efeitos de cada falha, a avaliação da severidade, ocorrência e detecção de cada falha, e o cálculo do Índice de Prioridade de Risco (RPN) para cada falha.

Uma vez que a FMEA do cenário "antes"esteja completa, a ferramenta é implementada e o processo ou sistema é monitorado por um período de tempo. Durante este período, quaisquer problemas ou falhas que ocorram são documentados e a FMEA é atualizada para refletir a nova situação.

Depois que a ferramenta estiver em uso por tempo suficiente para coletar dados significativos, a FMEA é realizada novamente para o cenário "depois". Esta FMEA irá refletir quaisquer mudanças que ocorreram como resultado da implementação da ferramenta.

Ao comparar o RPN do cenário "antes"com o RPN do cenário "depois", podemos quantificar o impacto dos grafos. Se o RPN do cenário "depois"for significativamente menor que o do cenário "antes", isso é um indicativo de que a ferramenta está tendo um impacto positivo. Por outro lado, se o RPN não mudar significativamente ou aumentar, isso pode indicar que a ferramenta não está tendo o impacto esperado e que mais investigação e ajustes podem ser necessários.

É importante ressaltar que este é um processo contínuo. A FMEA deve ser revisada e atualizada regularmente para refletir quaisquer mudanças no sistema ou processo, e para garantir que quaisquer melhorias sejam identificadas e implementadas.

3.6.2 Outras métricas e Indicadores-Chave de Desempenho (KPI - *Key Performance Indicators*

Para complementar a análise FMEA, outras ferramentas de gestão podem ser empregadas para avaliar a eficácia do novo sistema. Métricas como OEE, tempo de ciclo e FTY, descritas anteriormente na seção "Métricas de Gestão", são valiosas nesse contexto.

A OEE oferece uma visão abrangente da performance geral do processo. Seus indicadores refletem disponibilidade, rendimento e qualidade, elementos que podem ser otimizados significativamente com o uso do software de grafos.

A compreensão aprimorada que o software proporciona também pode refletir em menores tempos de ciclo, que indicam a duração total do processo, desde seu início até a conclusão. A capacidade de visualizar e analisar o fluxo de trabalho de forma abrangente pode ajudar a identificar gargalos e ineficiências que prolongam desnecessariamente o tempo de ciclo.

FTY, uma medida da eficácia do processo, é outro indicador que pode ser beneficiado. Essa métrica calcula a porcentagem de unidades produzidas que atendem

aos padrões de qualidade na primeira tentativa, sem a necessidade de retrabalho. Com a implementação do novo sistema, podemos esperar um aumento no FTY, revelando maior eficácia do processo como um todo.

FMEA, é possível ter uma visão mais completa do impacto da implementação do novo sistema no processo de calibração de motores.

Além disso, a integração dessas técnicas de gestão com um sistema de informação robusto, como o software de grafos, permite o monitoramento contínuo e a melhoria contínua dos processos. Isso facilita a identificação precoce de problemas, permitindo ações corretivas e preventivas para minimizar falhas e maximizar a eficiência do processo.

Em conclusão, a adoção de um sistema baseado em software de grafos, complementado por uma variedade de técnicas de gestão e métricas, pode trazer benefícios substanciais para a calibração de motores e processos similares. A análise profunda e a melhoria contínua resultante não apenas aumentam a eficiência, mas também melhoram a qualidade do processo, resultando em maior satisfação do cliente e sucesso a longo prazo.

4 Apresentação do protótipo

Depois de implementar a nova ferramenta baseada em grafos e definir os objetivos, é necessário avaliar o impacto que o sistema teve no processo de calibração de motores. Primeiramente este trabalho irá exibir as funcionalidades e aparência da ferramenta. Em seguida observa-se as métricas de gestão estabelecidas para fornecer uma avaliação quantitativa e qualitativa dos resultados. O intuito é não apenas validar a efetividade do sistema, mas também entender as possíveis áreas de melhoria e as lições aprendidas.

4.1 Descrição da ferramenta

A ferramenta pode ser dividida em duas partes distintas. A primeira parte será aquela com o uso diário e frequente pelos colaboradores da empresa. A segunda é uma página com direitos administrativos, esta será utilizada de uma forma limitada para criação de projetos e alterações no banco de dados.

4.1.1 Tela principal

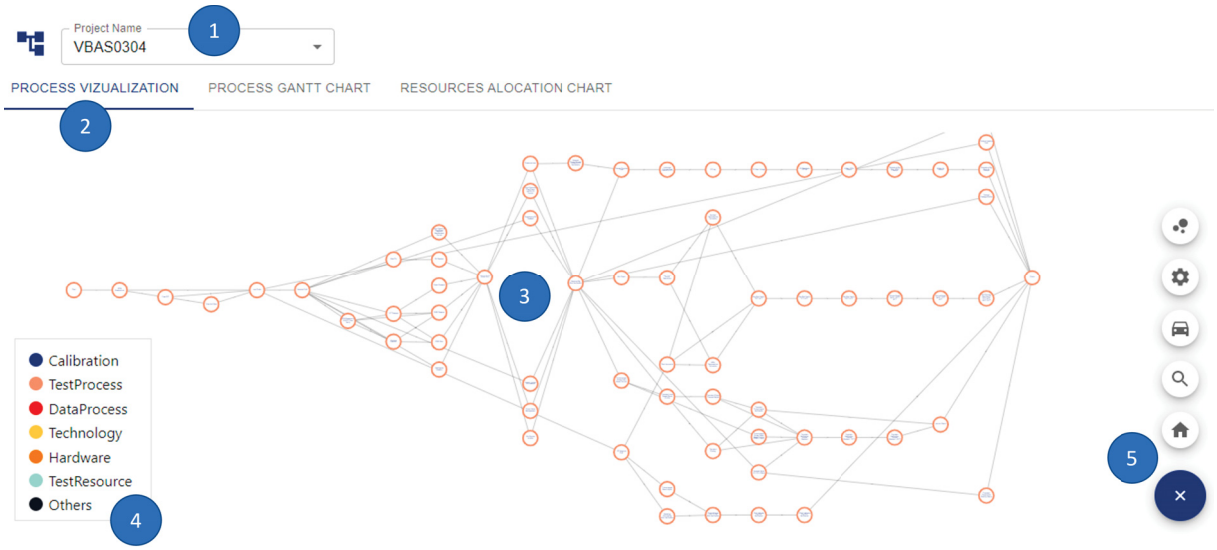

Figura 8 – Tela principal

A Figura 8 é a primeira tela com a qual o usuário se depara ao abrir a ferramenta. Nela, pode-se observar alguns pontos de atenção que estão enumerados. Eles são:

1. Caixa de seleção de projetos: Esta caixa busca todos os projetos cadastrados no banco de dados e disponibiliza para os usuários selecionarem e visualizarem o projeto de interesse.
2. Barra de guias: Esta barra permite o usuário navegar entre as três possiveis opções:
   a) Process Vizualization
   b) Process gantt chart
   c) Resources alocation chart
3. Janela de visualização: Esta janela é diferente para cada uma das guias da ferramenta e é um elemento canvas que exibe o elemento de interesse.
4. Legenda: Esta é uma legenda de cores que identifica os nós da tela
5. SpeedDial: Este botão exibe todas as possíveis ações que o usuario consegue efetuar na ferramenta

A seguir, é possível observar um melhor detalhamento de alguns destes pontos de interesse.

4.1.1.1 Janela de vizualização

Esta janela é um elemento canvas que poderá exibir três diferentes tipos de gráficos:

- Process Overview

  Quando selecionada esta opção, serão exibidos os links e nós retornados pelo banco de dados para a última busca especificada pelo usuário.

  A visuaização tem dois modos que são diretamente influenciados pela busca realizada, estes seriam: ordenado Figura 9, não ordenado Figura 10. O usuário irá selecionar o modo que deseja ao efetuar a busca.

  No modo ordenado, os nós são posicionados da esquerda para a direita, obedecendo o fluxo de processo a partir da utilização da biblioteca dagre.

  No segundo modo, os links e nós são posicionados na tela por uma simulação de força em que nós se repelem enquando links tentam aproximar aqueles dois nós que são relacionados.

- Process Gantt Chart

  Nesta guia será disponibilizado ao usuário o gráfico de Gantt gerado a partir do algoritmo CPM, Figura 11. Cada tarefa do processo recebe uma data estimada de execução e então é ordenada conforme o caminho crítico.

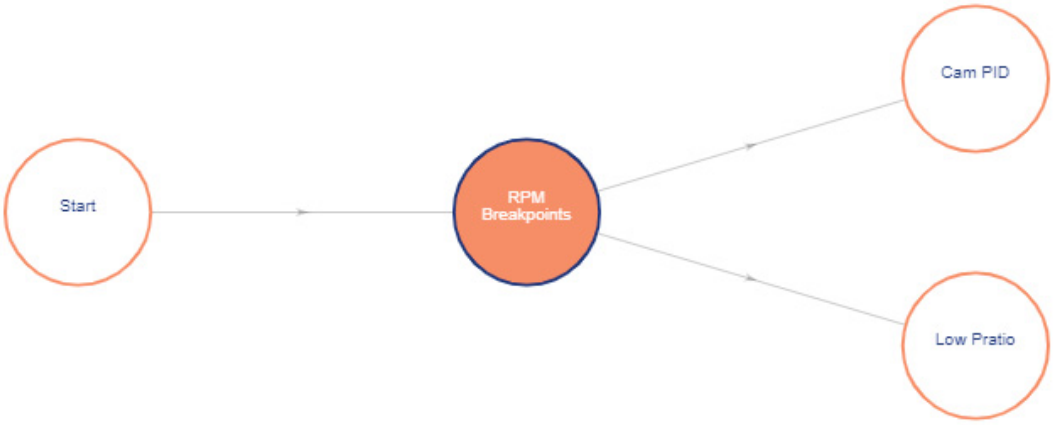

Figura 9 – Resultado da visualização no modo ordenado

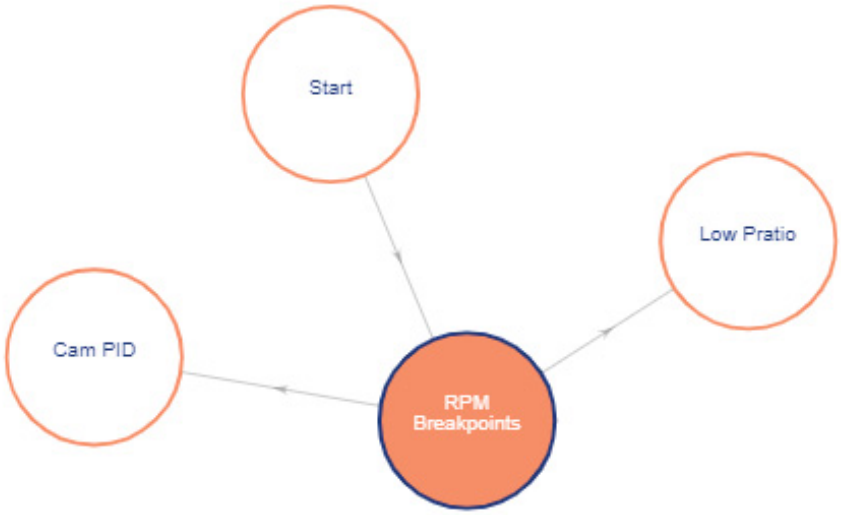

Figura 10 – Resultado da visualização no modo não ordenado

Uma mesma tarefa pode ter duas cores diferentes, sendo elas azul, que representa a duração total da tarefa, e laranja, que representa o *Slack* ou tempo disponível para a execução dessa tarefa.

Os marcadores verticais da cor laranja indicam a data prevista para que cada um dos *Quality Gates*, enquanto os marcadores em laranja devem indicar a situação atual de cada e se o projeto está ou não em atraso.

- Resources Allocation Chart

  O gráfico de barras mostrado na Figura 12 nos idica qual tarefa, a data e por quanto tempo um determinado recurso estará em uso ao longo do projeto. Esta

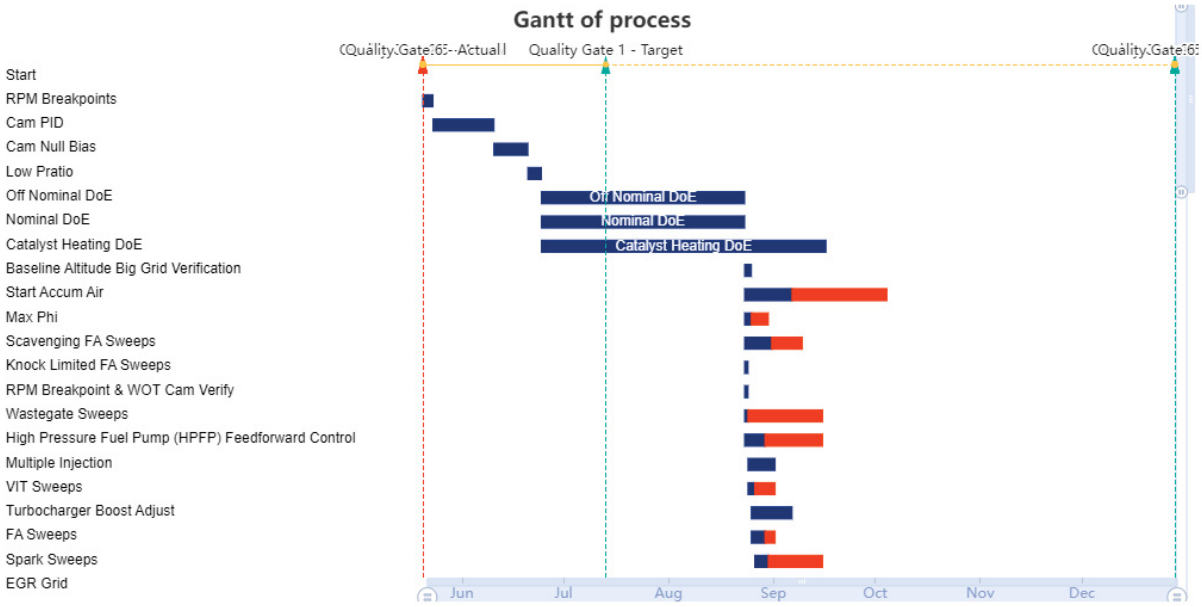

Figura 11 – Gráfico de gantt, gerado pelo do algoritmo CPM a partir da busca executada pelo usuário

tela busca facilitar a identificação de possíveis melhorias no processo e os impactos da escassez ou abundância de certo recurso na duração do mesmo.

4.1.1.2 *Speed Dial*

O botão *Speed dial* mostrado na Figura 13 é a principal interface de ações do usuário. Ele é um botão flutuante que, ao ser pressionado, revela múltiplas ações relacionadas que estavam ocultas. Os próximos tópicos detalham cada uma dessas ações.

- *Want to see something cool?*

  Esta ação não tem uma grande importância e funcionalidade ao usuário, porém é uma forma interessante de demostrar o tamnho e a complexidade de todo o processo de calibração de motores.

  A Figura 14 mostra a tela disponibilizada ao usuário quando o mesmo clica nesta opção. Nesta imagem podemos ver todos os links e nós presentes no banco de dados. Apesar de não ser uma visão intuitiva, ela serve para impactar o usuário e dar uma visão geral de quantas relações, tarefas e hardware existem no processo.

- *Configure resource availability*

  Esta opção permite ao usuário configurar o número de recursos que ele tem disponível para o determinado projeto no qual está trabalhando, bem como as datas alvo para cada um dos *Quality Gates* deste.

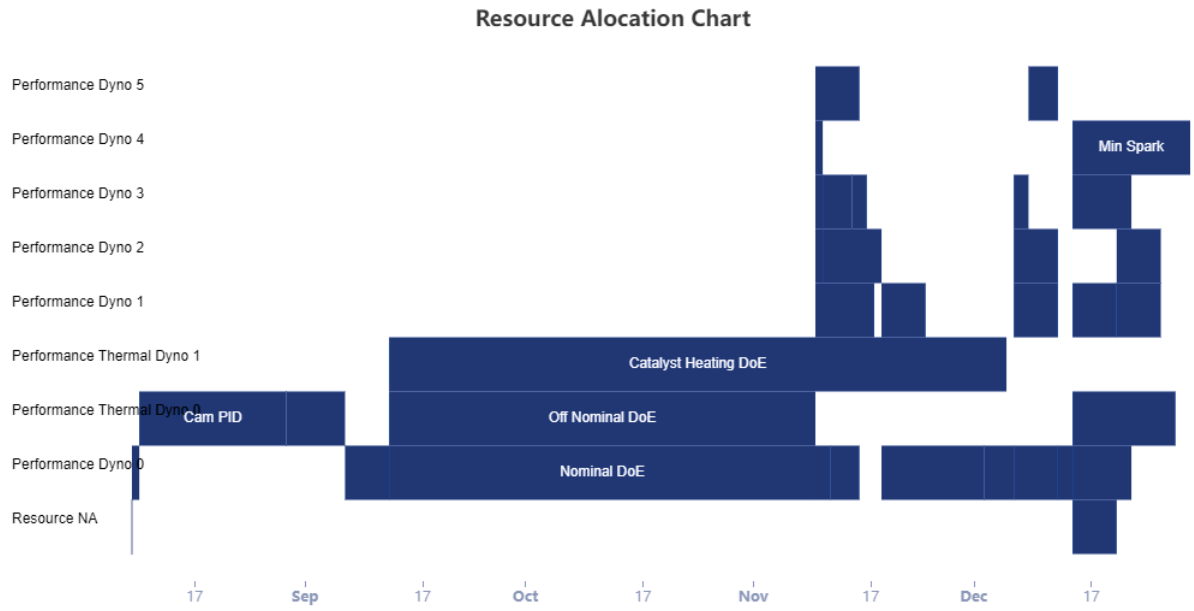

Figura 12 – Gráfico de barras indicando o tempo previsto em que cada um dos recursos estará em uso durante o projeto

A aba de configuração dos recursos, Figura 15, é composta por caixas de seleção em que o número máximo de recursos é estabelecido pela própria ferramenta e se baseia no número real de recursos disponíveis na fábrica.

Enquanto a aba de configuração dos *Quality Gates*, , Figura 16, é composta por caixa de seleção de datas em que o usuário pode escolher por meio de um calendário qual a data alvo para cada meta do projeto. A ideia por trás dos Quality Gates é garantir que cada fase de um projeto atenda a certos critérios de qualidade antes que o trabalho prossiga para a próxima fase. Eles são usados para identificar e corrigir problemas o mais cedo possível no ciclo de desenvolvimento, o que geralmente é mais eficiente e econômico do que corrigir problemas após a conclusão do projeto.

- *Filter by Engine*

  Este botão permite aos colaboradores abrirem uma janela *Select Engine* que, por sua vez, permite a seleção dos motores já cadastrados na ferramenta. A filtragem por estes motores têm duas principais funcionalidades, a primeira delas sendo exibir todos os processos envolvidos na calibração de cada um destes motores, Figura 18. Esta vista pode ser obtida ao selecionar o motor e clicar em *Get ordered processes*.

  Outra opção é clicar em *General Overview* que, por sua vez, irá exibir a Figura 19. Nesta vizualição é possível observar todos os hardwares que estão relacionados aos motores bem como os pontos de caibração que dependem de cada uma

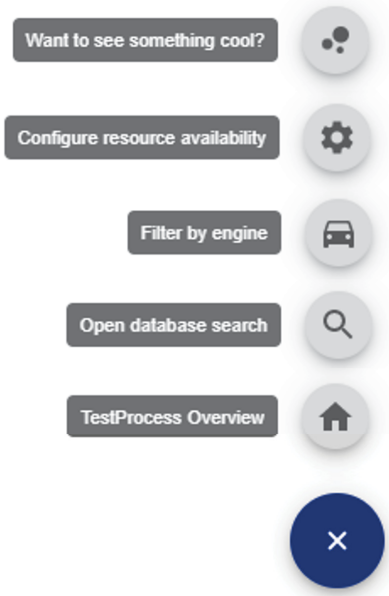

Figura 13 – Botão *Speed Dial* e todas suas opções

dessas peças. Isto possibilita uma visualização rápida de quais pontos devem ser recalibrados quando um componente do motor é modificado ou substituido de acordo com as neceessidades de cada veículo.

- *Search Database*

  Ao acionar o botão de busca, abre-se a tela responsável por requerer e definir os diversos parâmetros que são abordados para definição de uma *Query* ao banco de dados. Cada opção selecionada ou escrita nessa tela é traduzida em uma string que será inclusa da requisição enviada ao banco de dados.

  Esta tela possui alguns pontos de interesse que serão abordados em seguida, Figura 20.

  1. *What do you want to seach for?*: Esta seleção por *Checkboxes* identifica qual a *Label* do nó primário na requisição em Cypher.
  2. *How do you want it to be displayed?*: Ao escolher uma das duas opções o usuário estará decidindo qual será a ordenação do gráfico. Ordenado ou não ordenado, como já foi defindo ao longo deste texto.
  3. *Specify the name of the test process*: O primeiro campo opcional desta janela pode ser utilizado para buscar por um nó específico. Desta forma apenas nós relacionados ao nó principal serão retornados para visualização.
  4. *What package(s) are you searching for?*: Outro campo opcional que, por sua vez, disponibiliza um filtro de acordo com o *Package* do processo. Estas

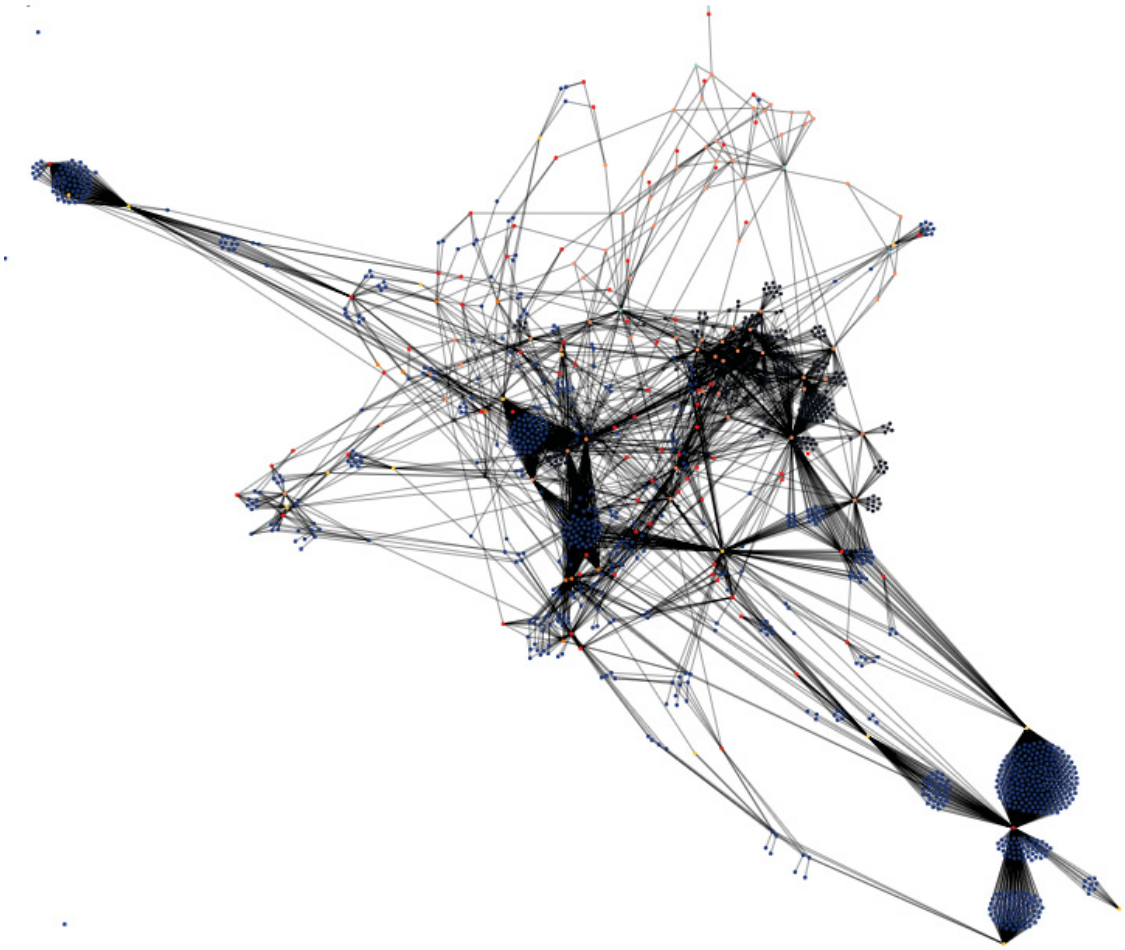

Figura 14 – Ilustração gráfica de todo o processo de calibração

propriedades são *label(s)* secundárias de cada um dos nós presentes no banco e segue uma caracterização interna da empresa.

5. *Relations*: As opções disponíveis para este campo são compatíveis com os possíveis tipos de relações disponíveis para a *label* principal do nó primário.

- *TestProcess Overview*

  Por fim, a primeira opção a aparecer quando se clica no *Speed Dial* é uma espécie de botão *Home*. Este botão dispara uma requisição padrão para o banco de dados e retorna à tela principal mostrada na 8

4.2 Aplicação das técnicas de gestão

Nesta seção, iremos aprofundar a análise do caso sob a ótica das técnicas de gestão e discutir como estas foram utilizadas para otimizar o processo de calibração de motores.

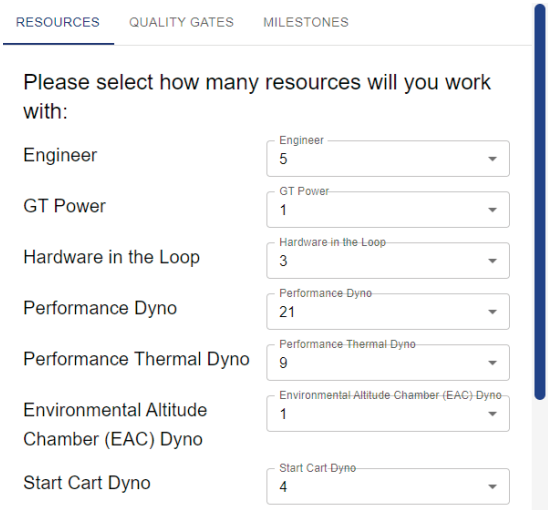

Figura 15 – Tela de configuração dos recursos disponíveis

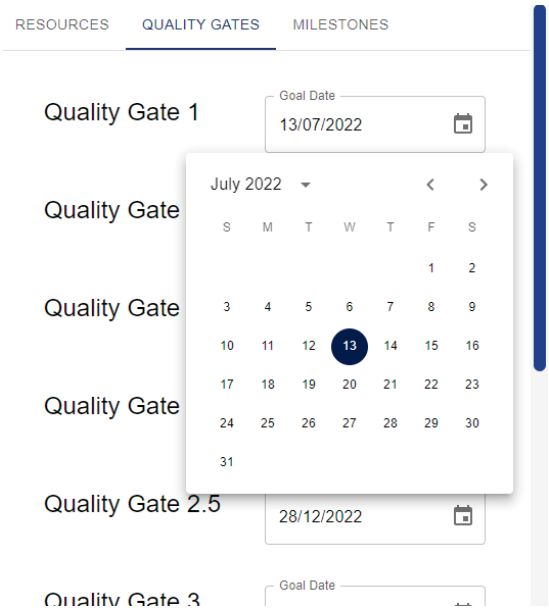

Figura 16 – Tela de configuração das datas alvo para os *Quality Gates*

Até a presente data deste texto, a empresa não utilizou a ferramenta Web de forma efetiva em seus projetos, portanto a análise de resultados será feita a partir do feedback dos times de gestão e engenharia, bem como os testes realizados durante o desenvolvimento.

4.2.1 *Failure Mode and Effect Analysis* - FMEA

Para análise FMEA, foram comparados os dois casos antes e após a implantação da ferramenta de grafos. Para executar essa tarefa, foram consultados diversos especialistas do time de Engenharia e, com isso, pode-se obter as Tabelas 2 e 3.

A análise da FMEA, comparando a planilha Excel com a aplicação Web revela

Figura 17 – Tela de seleção do motor desejado

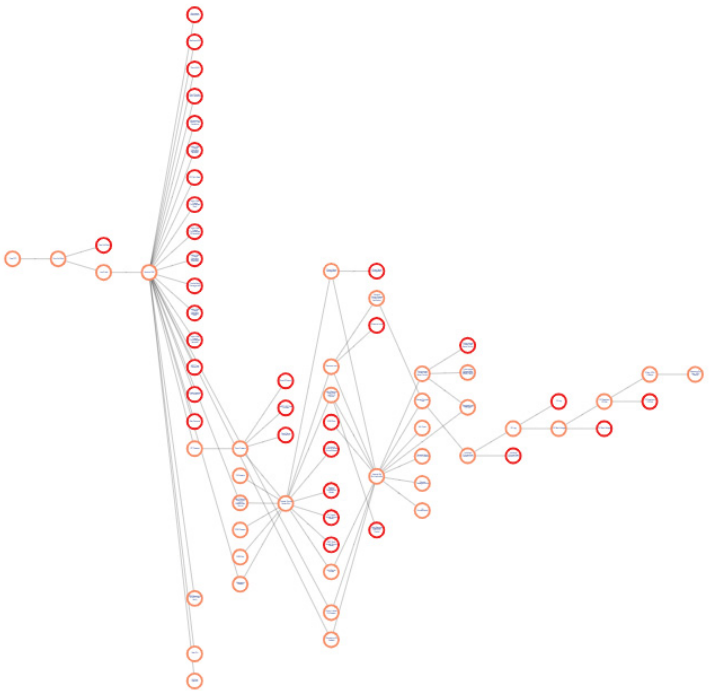

Figura 18 – Visão ordenada dos processos de calibração do motor selecionado

uma diferença significativa nos riscos associados a cada método, com a aplicação Web mostrando riscos substancialmente reduzidos em todas as áreas. Para executar a análise, reuniu-se um time de colaboradores que, após efetuarem testes com a nova ferramenta discutiram sobre os possíveis cenários.

Começando com a subutilização das células de dinamômetros, a ferramenta de aplicação Web oferece melhorias consideráveis. Enquanto o método da planilha Excel resultou em um RPN de 126, devido principalmente ao atraso das tarefas, a aplicação Web reduziu o RPN para 21, aprimorando a gestão do cronograma através de um gráfico de Gantt. Este gráfico permite uma visão clara e visual das tarefas e seus prazos, minimizando a probabilidade de atrasos e, por consequência, a subutilização dos dinamômetros.

Quando examinamos o potencial de erro na estimativa da duração do projeto, observamos outra melhoria dramática. O método da planilha Excel apresenta um RPN

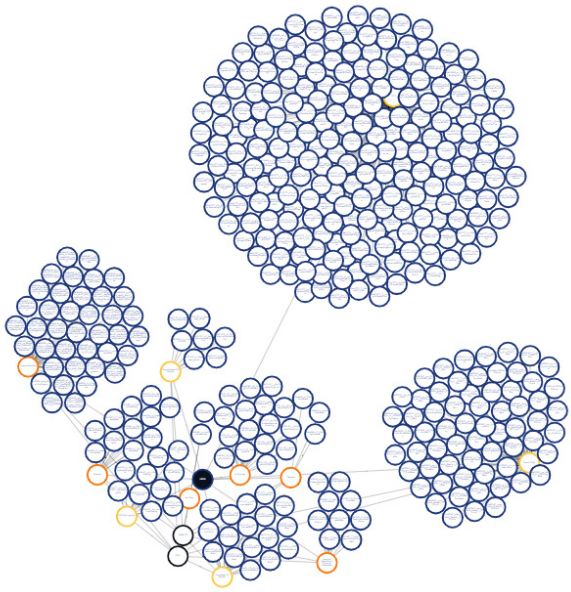

Figura 19 – Visualização dos componentes e pontos de calibração do motor selecionado

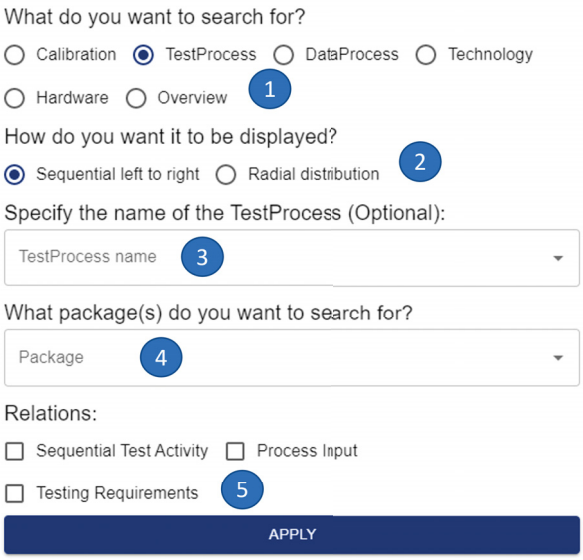

Figura 20 – Tela para busca de dados no servidor

de 144, mas a aplicação Web consegue reduzi-lo para 18. Isso ocorre porque a aplicação Web fornece uma visão mais precisa dos recursos disponíveis e permite uma melhor estimativa da duração das tarefas.

A questão da não identificação de possíveis caminhos paralelos também é notavelmente melhorada. Com a planilha Excel, o RPN é 288, enquanto com a aplicação Web é reduzido para apenas 4. A visualização aprimorada oferecida pela ferramenta permite ao gestor identificar e explorar possíveis caminhos paralelos para aumentar a eficiência.

Em relação à alocação ineficiente de novos recursos, a aplicação Web oferece um recurso de análise de diferentes cenários, reduzindo o RPN de 243 para 45. Isso permite ao gestor visualizar as consequências potenciais de várias ações antes de tomá-las, levando a decisões de alocação de recursos mais informadas e eficientes.

Na questão da não recalibração de pontos importantes, a aplicação Web também é superior. Com a planilha Excel, o RPN é 252, mas a aplicação Web reduz para 42. A aplicação permite uma busca mais eficiente do hardware substituído, o que facilita a recalibração.

Por fim, quando se trata de sub-alocação de recursos, a aplicação Web, com seu recurso de entrada de recursos disponíveis, diminui o RPN de 180 para 18, uma melhoria substancial comparada à planilha Excel.

Ferramenta de processo	Modo potêncial de falha	Efeito	S	Causa	O	Controle de prevenção	Controle de detecção	D	R
Planilha Excel	Controle do processo de calibração de motores	Sub-utilização das células de dinamômetros	7	Atraso de tarefas necessária à execução de novos testes	6		Goal posts	3	126
		Erro na estimativa de duração de um projeto	9	Estimativa de duração baseada em uma lista linear de tarefas	4		Identificação de atrasos frequentes	4	144
		Não identificação de possíveis caminhos paralelos	4	Visualização das tarefas em uma lista ordenada	8			9	288
		Alocação ineficiente de novos recursos	9	Adição de novos recursos na tentativa de acelerar o processo	3			9	243
		Não recalibração de pontos importantes	7	Troca de componentes do motor em meio ao processo de calibração	6		Falha da calibração em testes de controle	6	252
		Sub alocação de recursos	9	Alocação de recursos com base em conhecimento prévio	5		Atraso de processos	4	180

Tabela 2 – Tabela FMEA: Análise de riscos utilizando a tabela em excel para controle do processo

Ferramenta de processo	Modo potêncial de falha	Efeito	S	Causa	O	Controle de prevenção	Controle de detecção	D	R
Aplicação Web	Controle do processo de calibração de motores	Sub-utilização das células de dinamômetros	7	Falha do gestor no controle do cronograma	3		Acompanhamento do cronograma pelo gráfico de Gantt	1	21
		Erro na estimativa de duração de um projeto	9	Falha do gestor ao identificar os recursos disponíveis / Falha ao estimar a duração das tarefas no mapeamento do processo	2		Indisponibiliade de recursos previamente considerados Identificação de atrasos frequentes	1	18
		Não identificação de possíveis caminhos paralelos	4	Falha do gestor	1		Visualização do caminho por meio da ferramenta	1	4
		Alocação ineficiente de novos recursos	9	Falha do gestor	1		Análise de difrentes cenários pelo resource allocation chart	5	45
		Não recalibração de pontos importantes	7	Falha da equipe técnica	3		Busca do hardware substituido na ferramenta	2	42
		Sub alocação de recursos	9	Falha do gestor	2		Input dos recursos disponíveis na configuração da ferramenta	1	18

Tabela 3 – Tabela FMEA: Análise de riscos utilizando a aplicação web para controle de processos

4.2.2 OEE: Aumentar a Eficiência Global dos Equipamentos

Para analisar a eficiência global das células de dinamômetro, utilizáva-se os gráficos *Goal Post*. Porém os meios de controle para identificar possíveis tempos de inutilização das ferramentas devido à tarefas em atraso era limitado. Devido à complexidade da planilha e à falta de comunicação entre gestores e o time de engenharia, a possibilidade de identificar essas situação préviamente era baixa.

A ferramenta web trouxe a possibilidade de tratar tarefas similarmente ao método *Kanban*. Portanto, com a maior interação do time e visualização das tarefas pelo gestor, espera-se uma melhoria bastante significativa quanto à antecipação do time a situações deste tipo.

4.2.3 Tempo de Ciclo: Reduzir a Duração do Processo

A redução do tempo de ciclo para cada um dos projetos poderá ser reduzida a partir da maior paralelização do processo. Hoje, a planilha em excel atoa como um gargalo para o gestor, pois exibe uma lista ordenada e limita a visão do processo como ele realmente é.

Paralelizando de forma efetiva o projeto, será possível acelerar sua eecução e atingir metas sem, necssariamente, utilizar mais recursos. A ferramenta contribui com este fator ao mostrar de forma gráfica todos os possíveis caminhos e interdependências das tarefas dentro de cada projeto.

4.2.4 FTY: Aumentar a Eficiência do Processo

O aumento da eficiência de processo devido ao uso da ferramenta web poderá se manifestar de algumas formas. Primeiramente, reduzindo o tempo ocioso das células. Mas também mostrando ao gestor que a adição de mais uma célula ao processo não será de grande valia.

Processos em atraso ganham grande atenção e sofrem com enormes pressões de toda a cadeia gerencial e produtiva. Portanto, é comum que gestores solicitem e utilizem cada vez mais recursos para atigir suas métas. Contudo nem sempre a adição de mais recursos irá efetivamente acelerar a execução de um projeto, mas irá impactar efetivamente nos custos e eficiência deste.

Com a análise de diferentes situações de disponibilidade de recursos, por meio da ferramenta web, espera-se que o gestor faça sua escolha tendo à sua disposição todos os dados necessários. Assim, podendo argumentar de forma educada se a dição ou não de mais recursos será a solução para seu projeto.

5 Conclusão

O desenvolvimento desta monografia foi guiado pelo objetivo principal de implementar uma ferramenta que empregasse grafos como meio para aprimorar o processo de calibração de motores. O intuito era alcançar maior eficiência e qualidade nesse procedimento, que é crucial na indústria automotiva.

Em relação aos objetivos específicos estabelecidos no início do trabalho, podemos concluir o seguinte:

- **Melhoria dos indicadores de desempenho:** A aplicação de grafos demonstrou ser uma ferramenta valiosa na otimização dos indicadores, refletindo em melhores práticas e métodos durante a calibração.
- **Redução do tempo de ciclo:** Através da organização proporcionada pelo modelo baseado em grafos, foi possível mapear e eliminar gargalos, resultando em uma redução significativa no tempo de ciclo do processo de calibração.
- **Aumento da eficiência do processo:** Com a capacidade de visualizar e analisar interações complexas entre variáveis, a ferramenta permitiu uma abordagem mais sistemática, aumentando a eficiência geral do processo de calibração.
- **Melhor visualização do uso de recursos:** A representação gráfica proporcionada pela ferramenta não só facilitou a compreensão do processo como um todo, mas também ajudou na alocação e gerenciamento de recursos, garantindo que fossem utilizados de maneira ótima.

Portanto, é evidente que a abordagem proposta nesta monografia não só atendeu aos objetivos estabelecidos, mas também trouxe inovações significativas para o campo da calibração de motores. A incorporação de técnicas baseadas em grafos no processo tradicional de calibração provou ser um avanço promissor, abrindo portas para futuras pesquisas e melhorias contínuas no domínio.

5.1 Trabalhos futuros

Este trabalho representa apenas o começo da integração entre teorias de grafos e o processo de calibração de motores. Embora tenhamos alcançado avanços significativos, existem várias direções a serem exploradas para aprimorar e expandir os resultados obtidos. Algumas propostas para trabalhos futuros incluem:

- **Modelos Dinâmicos:** A ferramenta atualmente trabalha com grafos estáticos. Um desenvolvimento futuro poderia incluir a implementação de grafos dinâmicos que se ajustam e mudam em tempo real com base em dados de sensores e feedbacks durante a calibração.
- **Integração com Inteligência Artificial:** A ferramenta pode ser potencializada com algoritmos de aprendizado de máquina. Isso permitirá a predição e a adaptação a possíveis falhas ou ineficiências, tornando o processo mais robusto e autônomo.
- **Interface de Usuário Aprimorada:** Embora a visualização baseada em grafos seja um avanço, a interface do usuário pode ser continuamente melhorada para ser mais intuitiva, incorporando visualizações 3D ou realidade aumentada para facilitar a interpretação.
- **Expansão para Outras Áreas:** Os princípios de modelagem e otimização aplicados na calibração de motores podem ser adaptados para outras áreas da indústria automotiva, como teste de sistemas de freios ou otimização de aerodinâmica.
- **Integração com Outras Tecnologias:** A combinação da ferramenta com outras tecnologias emergentes, como Internet das Coisas (IoT) ou sistemas de monitoramento avançado, pode proporcionar um controle mais granular e feedback em tempo real durante o processo de calibração.
- **Estudos de Caso e Testes em Ambientes Reais:** Será fundamental testar a ferramenta em uma variedade maior de cenários e ambientes reais para garantir sua robustez e eficácia em diferentes situações e com diferentes tipos de motores.

Assim, vemos um horizonte vasto e promissor à frente. O trabalho atual semeou as primeiras ideias, mas a verdadeira revolução na calibração de motores baseada em grafos está apenas começando.

Referências

AGENCY, I. E. Energy technology perspectives 2020. IEA, 2020. Citado na página 1.

BAIDU, I. **Echarts**. 2021. <https://echarts.apache.org/en/index.html>. Citado na página 22.

BAIDU, I. **Echarts - Network Graph**. 2021. <https://echarts.apache.org/examples/en/editor.html?c=graph>. Acesso em 6 de maio de 2023. Citado na página 22.

BOSTOCK, M. **D3.js - Data-Driven Documents**. 2021. <https://d3js.org/>. Citado na página 21.

CANTELON, M.; HOLOWAYCHUK, T.; RAJLICH, N. **Node.js in Action**. [S.l.]: Manning Publications Co., 2013. Citado na página 21.

CINPOERU, M.; GHIRAN, A.-M.; HARKAI, A.; BUCHMANN, R. A.; KARAGIANNIS, D. Model-driven context configuration in business process management systems: An approach based on knowledge graphs. In: PAŃKOWSKA, M.; SANDKUHL, K. (Ed.). **Perspectives in Business Informatics Research**. Cham: Springer International Publishing, 2019. p. 189–203. Citado na página 17.

COMMISSION, E. **Regulation (EU) 2018/1832 of the European Parliament and of the Council of 5 November 2018 amending Regulation (EC) No 715/2007 on type approval of motor vehicles with respect to emissions from light passenger and commercial vehicles (Euro 5 and Euro 6) and on access to vehicle repair and maintenance information**. [S.l.], 2018. Citado 2 vezes nas páginas 1 e 9.

CORMEN, T.; LEISERSON, C.; RIVEST, R.; STEIN, C. **Introduction to algorithms**. [S.l.]: MIT press, 2009. Citado na página 11.

CYPHER Query Language. Accessed: 2023-05-06. Disponível em: <https://neo4j.com/developer/cypher/>. Citado na página 17.

DAI, Y.; ZHANG, Y.; LI, Y.; WANG, X. Application of graph theory in vehicle assembly network planning. **International Journal of Production Research**, Taylor & Francis, v. 56, n. 12, p. 4253–4266, 2018. Citado na página 16.

DEB, K. **Multi-objective optimization using evolutionary algorithms**. [S.l.]: John Wiley & Sons, 2001. Citado na página 8.

DEB, K.; DEB, D. **Optimization for Engineering Design: Algorithms and Examples**. [S.l.]: PHI Learning Pvt. Ltd., 2011. Citado na página 9.

DIESTEL, R. **Graph Theory**. 5. ed. [S.l.]: Springer, 2017. Citado na página 10.

DIJKSTRA, E. A note on two problems in connexion with graphs. **Numerische mathematik**, Springer, v. 1, n. 1, 1959. Citado 2 vezes nas páginas 11 e 13.

FENG, S.; LI, X.; WANG, X.; LIU, Y.; WANG, J.; TAM, L. M. A review of recent advances and research on internal combustion engine waste heat recovery. **Applied Thermal Engineering**, Elsevier, v. 128, p. 837–853, 2018. Citado na página 2.

FLANAGAN, D. **JavaScript: The Definitive Guide**. 7. ed. [S.l.]: O'Reilly Media, Inc., 2020. Citado na página 20.

GUILLAUME, P.; CHARLET, A. Model-based calibration: an application in automotive engine control. **IFAC Proceedings Volumes**, Elsevier, v. 43, n. 14, p. 349–354, 2010. Citado na página 7.

HEYWOOD, J. B. **Internal Combustion Engine Fundamentals**. [S.l.]: McGraw-Hill, 1988. Citado 2 vezes nas páginas 1 e 7.

JIN, X.; GUZZELLA, L.; ONDER, C. Machine learning techniques for engine calibration and control. **Applied Energy**, Elsevier, v. 226, p. 983–992, 2018. Citado na página 7.

KERZNER, H. **Project Management: A Systems Approach to Planning, Scheduling, and Controlling**. 12. ed. [S.l.]: Wiley, 2017. Citado na página 1.

KERZNER, H. **Project Management: A Systems Approach to Planning, Scheduling, and Controlling**. 12. ed. [S.l.]: John Wiley & Sons, 2017. Citado 2 vezes nas páginas 12 e 16.

KIM, K. H.; PARK, M. J. A new heuristic for resource-constrained project scheduling. **Information Sciences**, v. 177, n. 18, p. 3665–3682, 2007. Citado na página 16.

KLIJN, E. L.; MANNHARDT, F.; FAHLAND, D. Classifying and detecting task executions and routines in processes using event graphs. In: POLYVYANYY, A.; WYNN, M. T.; LOOY, A. V.; REICHERT, M. (Ed.). **Business Process Management Forum**. Cham: Springer International Publishing, 2021. p. 212–229. ISBN 978-3-030-85440-9. Citado na página 16.

KOSMACZEWSKI, A. **Mobile JavaScript Application Development**. [S.l.]: O'Reilly Media, Inc., 2012. Citado na página 20.

MORTON, T. E.; PENTICO, D. W. **Heuristic Scheduling Systems: With Applications to Production Systems and Project Management**. [S.l.]: John Wiley & Sons, 2014. Citado na página 15.

NEO4J. **Neo4j Developer Resources**. 2023. <https://neo4j.com/developer/>. Accessed: 2023-05-06. Citado na página 19.

NEO4J: The World's Leading Graph Database. Accessed: 2023-05-06. Disponível em: <https://neo4j.com/>. Citado na página 17.

NEWMAN, M. E. **Networks: An Introduction**. [S.l.]: Oxford University Press, 2018. Citado 2 vezes nas páginas 11 e 16.

PAL, A.; ZHU, L.; WANG, Y.; ZHU, G. Data-driven model-based calibration for optimizing electrically boosted diesel engine performance. **International Journal of Engine Research**, v. 24, n. 4, p. 1515–1529, 2023. Citado na página 9.

PETEGHEM, V. V.; VANHOUCKE, M. State of the art exact and heuristic solution procedures for the resource-constrained project scheduling problem. **European Journal of Operational Research**, v. 184, n. 3, p. 807–827, 2008. Citado na página 16.

PUNDIR, B. P. **Advanced Engine Technology**. [S.l.]: Firewall Media, 2016. Citado na página 9.

ROBINSON, I.; WEBBER, J.; EIFREM, E. **Graph Databases: New Opportunities for Connected Data**. [S.l.]: O'Reilly Media, Inc., 2013. Citado na página 17.

SILVA, V. A. **ForceGraph2D**. 2021. <https://github.com/vasturiano/force-graph>. Citado na página 21.

STAMATIS, D. **Failure Mode and Effect Analysis: FMEA from Theory to Execution**. ASQ Quality Press, 2003. ISBN 9780873895989. Disponível em: <https://books.google.com.br/books?id=Sl25A29t1E8C>. Citado na página 27.

STJERNSTRÖM, B.; HOLMSTRÖM, J. Design and implementation of a flexible material flow control system based on graph theory and heuristics. In: IEEE. **Proceedings of the 36th Annual Hawaii International Conference on System Sciences**. [S.l.], 2003. p. 9–pp. Citado na página 16.

STONE, R. **Introduction to Internal Combustion Engines**. [S.l.]: Macmillan, 1999. Citado na página 7.

TILKOV, S.; VINOSKI, S. Node.js: Using javascript to build high-performance network programs. **IEEE Internet Computing**, IEEE, v. 14, n. 6, p. 80–83, 2010. Citado na página 20.

WEST, D. B. **Introduction to Graph Theory**. 2. ed. [S.l.]: Prentice Hall, 2001. Citado na página 9.

WORKFLOW simulation for operational decision support using event graph through process mining. **Decision Support Systems**, v. 52, n. 3, p. 685–697, 2012. ISSN 0167-9236. Citado na página 17.