**DESENVOLVIMENTO DE SOFTWARE PARA CÁLCULOS ANALÍTICOS APLICADO À BAJA SAE**

Bruno Mello Silveira

Emerson Marins Marques Souvain

Marcus Vinícius Mariath Machado Nogueira

Resumo

O presente estudo visa apresentar o desenvolvimento de uma Web API utilizando a linguagem de programação C# capaz de automatizar etapas de cálculos analíticos feitas manualmente no desenvolvimento de um projeto de veículo off-road Baja SAE. A primeira versão do projeto é capaz de realizar cálculos das reações e tensões para os componentes do sistema de suspensão considerando equilíbrio estático e este atuando como uma treliça. Com a automatização dos cálculos analíticos foi possível considerar a influência de mais fatores, aumentar a confiabilidade estrutural do veículo off-road, podendo identificar erros com maior facilidade, maior agilidade nos cálculos e menor esforço. Além de ter sido aplicado com sucesso pela equipe Mud Runner Baja SAE do CEFET/RJ.

Palavras-chave: cálculos analíticos, análise estrutural, automatização, Baja SAE, API Web.

Abstract

This study aims to present the development of a Web API using the C# programming language capable of automating manual analytical calculation steps in the development of a Baja SAE off-road vehicle project. The first version of the project can perform calculations of reactions and stresses for the components of the suspension system considering static equilibrium and this one acting as a truss. The automation of analytical calculations allows considering the influence of more factors, increasing the structural reliability of the off-road vehicle, being able to identify errors easier, greater flexibility in calculations and less effort. Moreover, it was successfully applied to the Mud Runner Baja SAE team from CEFET/RJ.

Keywords: analytical calculations, structural analysis, automatization, Baja SAE, API Web.

**1 INTRODUÇÃO**

O programa Baja SAE, apresentado por (SAE BRASIL, 2021), oferece aos estudantes de engenharia e cursos relacionados a oportunidade de aplicar na prática os conhecimentos obtidos durante a graduação, visando desenvolver profissionais mais capacitados ao mercado de trabalho. Nesse programa, os estudantes devem formar equipes que representarão a instituição de ensino superior na qual estudam e realizam o desenvolvimento de um veículo off-road*,* além de participarem de competições que promovem uma avaliação comparativa entre os projetos.

No projeto, faz-se necessário calcular e validar estruturalmente os componentes do protótipo desenvolvido pela equipe de Baja SAE, com o principal objetivo de assegurar a confiabilidade e garantir que não apresentarão falhas durante o funcionamento. Nesse contexto, comumente encontram-se desafios e problemáticas com alto grau de complexidade que são mais bem solucionados com o uso de recursos computacionais. Pensando nisso, o presente trabalho busca apresentar o desenvolvimento de um software nos moldes de uma API Web em linguagem de programação C# capaz de automatizar as etapas de cálculos analíticos realizados por uma equipe de Baja SAE, tomando como base a equipe Mud Runner do CEFET/RJ. Com isso, será possível encontrar condições de contorno mais precisas para serem aplicadas nas análises de elementos finitos. Além disso, será possível disponibilizá-la na internet, com o intuito de auxiliar demais equipes no desenvolvimento de seus projetos, e permitir integrações entre sistemas, por exemplo a API apresentada neste estudo com outras externas, tanto softwares desenvolvidos por outros estudantes ou softwares comumente utilizados por equipes de competição, como Ansys, SolidWorks, entre outros.

**1.1 REVISÃO BIBLIOGRÁFICA**

O Projeto Baja Sae tem uma influência positiva na formação acadêmica de alunos de engenharia, é o que afirma (FERREIRA, 2011) em seu estudo envolvendo a evolução do conhecimento do estudante ao participar do Projeto Baja Sae. Nesse sentido, foi verificado habilidades como capacidade de trabalho em equipe, capacidade de comunicação escrita e oral, capacidade de liderança e analítica, entre outros. Além disso, o desenvolvimento de um projeto possibilita ao universitário aplicar os conceitos teóricos vistos em sala de aula, preparando-o para o mercado de trabalho. Ademais, conforme mostrado por (REIS, 2003), o projeto Baja SAE também propicia melhorias à sociedade, pois em seu estudo foi apresentado a utilização de LEGO e programação para construir uma miniatura de um protótipo de Baja SAE para motivar crianças e adolescentes a aprender e desenvolver suas ideias.

No desenvolvimento de um projeto Baja Sae, o sistema de suspensão possui uma grande importância por ser necessário absorver impactos com grande frequência e conferir boa manobrabilidade e dinâmica veicular em meio a terrenos irregulares. (MINOZZO, 2012) apresentou esta importância, argumentando sobre a necessidade de conhecer os esforços dinâmicos atuantes no sistema de suspensão para projetar adequadamente os componentes. (FERNANDES, 2015) fez uma revisão dos conceitos importantes aplicados a suspensão e analisou os parâmetros utilizados no sistema de suspensão através do software Adams Car.

Atualmente, pode-se observar que a automação de processos possui grande importância em diversos trabalhos por buscar a melhor performance de uma determinada tarefa, conciliando menor custo, alta produtividade e minimização de falhas. (SENA, 2019) utilizou programação orientada a objetos juntamente à linguagem C# para desenvolver um software que automatiza processos de projetos arquitetônicos através da integração com a API do Revit.

Devido a seus benefícios, a programação vem ganhando espaço entre as equipes de Baja SAE, as aplicações possíveis compõem uma grande gama de possibilidades e podem trazer resultados satisfatórios em competições nacionais e internacionais. (DRUNN, 2018) apresentou o desenvolvimento de um sistema de transmissão de câmbio automático utilizando Arduino, tendo como objetivo encontrar rotações ideais para as trocas de marchas, de forma que o máximo torque seja transmitido as rodas e possibilite maior desempenho do veículo em competições. (SCHOMMER, 2021) desenvolveu um software utilizando linguagem de programação Python para calcular matricialmente as reações estáticas no sistema de suspensão de um veículo de Baja ou Fórmula SAE quando este é submetido a uma força e um momento aplicados no pneu. Além disso, ele explicou que estes resultados podem ser aplicados diretamente em análises de elementos finitos em softwares, como Ansys, para que uma análise estrutural completa possa ser feita. Vale salientar que este último estudo serviu como a principal base para o desenvolvimento da primeira funcionalidade do software desenvolvido neste trabalho.

**1.2 CONCEITOS FUNDAMENTAIS DE BAJA SAE**

Neste tópico, diversos conceitos acerca do projeto de Baja SAE serão explicados para uma melhor compreensão. O primeiro é o sistema de suspensão que é responsável pela dinâmica veicular e tem a função de absorver os impactos sofridos pelo automóvel e de garantir a estabilidade do veículo em diferentes terrenos. Para o Baja, a suspensão é considerada uma prioridade de projeto, pois o carro é submetido a provas de resistência em terrenos off-road com buracos, ladeiras, lamas e pedras.

O conjunto de suspensão analisado é constituído pelos seguintes componentes: balanças superiores e inferiores, amortecedores, braço de convergência (no caso da traseira) e braço de direção (no caso da dianteira). Para realizar os cálculos nesses componentes pode-se considerar o sistema de suspensão atuando como uma treliça, que consiste em um conjunto de barras articuladas entre si cujas reações são no eixo de cada componente. Esta consideração permite simplificar os cálculos analíticos, já que a direção dos vetores de força é conhecida, tendo somente seu valor como incógnita.

No campo de resistência dos materiais os conceitos de tensão, flambagem e fadiga dividem as atenções. A tensão representa o estudo da distribuição das forças ao longo de uma área, logo é de suma importância entender como os esforços se comportam em determinado corpo, para a correta escolha do material e melhores decisões de otimização. Nesse presente estudo, são utilizados os conceitos de tensões normal, fletora, cisalhamento e equivalente, que serão melhores apresentados com definição e equação utilizadas mais a diante.

A falha do material no cenário do Baja SAE comumente é causada por conta dos esforços repetitivos que os componentes do veículo são submetidos durante o percurso da competição. Esses carregamentos cíclicos aplicados aos componentes podem ocasionar uma falha no componente denominada falha por fadiga, que começa através de uma trinca e dependendo do número de ciclos pode se propagar até uma fratura. Dessa forma, é importante fazer análises estruturais dos componentes para encontrar e visualizar os valores e comportamentos das tensões atuantes, para a partir disso realizar uma análise considerando o número de ciclos de fadiga escolhido para o componente, além de descobrir o fator de segurança e a vida em fadiga da peça analisada.

Nesse contexto, existem três formas de obter os valores de tensões atuantes nos componentes de um veículo Baja SAE. A primeira forma é realizando uma análise analítica que consiste em uma análise estrutural feita através de cálculos manuais. A segunda é uma análise experimental que é realizada através de ensaios com sistemas de captação de dados no protótipo real. Já a outra opção é uma análise numérica que consiste em uma simulação computacional desenvolvida em um protótipo virtual do componente, dessa forma é possível visualizar o comportamento das tensões atuantes através de uma representação virtual próxima do real. Para realização de uma análise numérica, é utilizado o método dos elementos finitos, que consiste em dividir o componente em pequenas parcelas. Segundo (MIRLISENNA, 2016), essas parcelas são denominadas como elementos e são criadas para permitir o computador resolver com mais eficiência um problema complexo, subdividindo-o em pequenos problemas mais simples e consequentemente aumentando a precisão dos resultados das análises estruturais.

Por último, outro conceito de suma importância é o de flambagem, que é o fenômeno que ocorre quando o componente é submetido a um esforço axial no eixo da peça, e apresenta deformações significativas em torno dele. Esse comportamento pode representar um perigo para estrutura, pois com o desvio da posição natural da peça, ao longo do eixo, ocorre um incremento nos esforços de flexão, que podem ocasionar na falha da estrutura. Em suma, o fenômeno de flambagem deve ser evitado, pois pode acarretar num panorama de falha iminente.

**1.3 CONCEITOS FUNDAMENTAIS DE PROGRAMAÇÃO**

Neste tópico, alguns conceitos da área de ciência da computação serão explicados para uma melhor compreensão deste estudo. A linguagem de programação C#, escolhida para este trabalho, é, segundo (MICROSOFT, 2021), “uma linguagem de programação moderna, orientada a objeto e de tipo seguro”. Além disso, é muito utilizada no setor de tecnologia seu poder computacional, além de possuir uma vasta documentação ofertada gratuitamente pela *Microsoft*.

O próximo conceito é a Web API, que de acordo com (NAVARRO, 2012) é uma API, sigla para *Application Programming* *Interface*, que expõe suas funcionalidades na internet. Segundo (MICROSOFT, 2019), uma API “específica como os componentes e sistemas de software devem interagir uns com os outros” e, segundo (TECHTUDO, 2020), representa um conjunto de padrões e protocolos que possibilitam a integração entre sistemas. Posto isto, foi utilizado o padrão SOLID, que apresenta princípios de boas práticas de programação orientada a objeto (POO) com o objetivo de tornar o código limpo, de fácil manutenção e escalável, e o protocolo HTTP, que expõe as estruturas de comunicação Web necessárias.

Por utilizar protocolo HTTP, para que o software funcione corretamente faz-se necessário utilizar alguns recursos como entrada e saída de dados na formatação JSON, chamadas HTTP por meio de *endpoint* e estrutura de criação de URLs estáticas, sendo escolhido o *framework* ASP.NET MVC (RICK ANDERSON, 2019) e (GASPAROTTO, 2014).

Outro conceito importante é *Swagger*, que de acordo com (SMARTBEAR, 2021) é uma ferramenta open source para desenvolvimento de APIs que facilita no design, construção, documentação, teste e padronização, e permite ao usuário acessar os *endpoints* disponíveis através de uma interface simples e de fácil compreensão com documentações, exemplos e explicações sobre os recursos, conforme pode ser visto na Fig. 1.

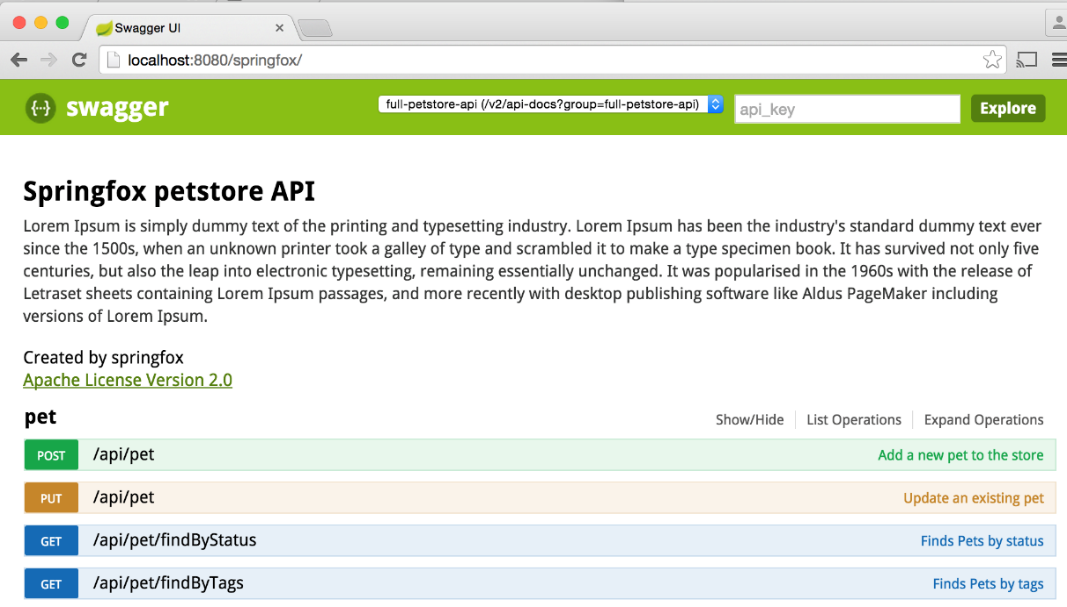


Figura 1 – *Swagger* da API desenvolvida para este estudo (STACKER OVERFLOW, 2018)

Com o objetivo de tornar as operações mais rápidas e otimizadas, foi utilizado o conceito de paralelismo e assincronismo através da classe *Task* (TASK CLASSE, 2021), nativa do C#. Paralelismo diz respeito a execução de tarefas em paralelo, ou seja, mais de uma tarefa será executada ao mesmo tempo. Assincronismo representa a execução de tarefas em *threads* separadas, isto é, são feitas em contextos de processamento diferentes. Mesmo apresentando reduções consideráveis no tempo de execução do código, assincronismo e paralelismo são recursos que devem ser utilizados com cautela, uma vez que fazem com que mais recursos computacionais sejam utilizados. Dessa forma, é necessário fazer uma análise de custo-benefício levando em consideração a quantidade de tarefas em paralelo ou assíncronas e dos recursos computacionais disponíveis, principalmente processador e memória RAM.

O último conceito abordado é o de herança, tópico intrínseco à programação orientada a objetos. Isto é utilizado em todo o código para evitar replicações de código e permitir a criação de abstrações, tornando algumas classes mais genéricas e impondo que alguns comportamentos sejam implementados em determinados contextos. De acordo com (DE CAMARGO, 2010), “a herança possibilita que as classes compartilhem seus atributos, métodos e outros membros da classe entre si”, ou seja, definindo que uma classe herda da outra, a classe filho ou classe derivada recebe as características da classe pai ou classe base.

**2 IMPLEMENTAÇÃO NUMÉRICA**

Com o objetivo de facilitar a implementação de novos recursos e evitar repetições de código, foram criadas classes base para os contratos, que são todos as classes que definem os dados de entrada e saída de cada funcionalidade ou operação, e as operações, que são classes que possuem alguma funcionalidade que é exposta para o usuário, sendo necessário somente o desenvolvimento das lógicas e contratos específicos. A classe base para os dados de saída possui as seguintes propriedades: *Success*, de tipo booleano, sendo verdadeiro, quando a operação executar corretamente, e falso, caso falhe; *HttpStatusCode*, que contém o status HTTP da operação, podendo ser 200, em casos de sucesso, 400, em casos de erro na validação, entre outros; *Errors*, lista que contém os erros que ocorrem durante a execução; e *Data*, que contém os dados principais da resposta, sendo específica para cada operação baseando-se nas diferentes necessidades do usuário. Além de possuir métodos que permitem manipulá-la adicionando erro e/ou alterando o status HTTP. A classe base para todas as operações determina o fluxo que deve ser seguido, em que primeiro os dados de entrada são validados dentro do método *ValidateOperationAsync*, caso sejam válidos, segue para o método *ProcessOperationAsync*, em que a operação será de fato executada e, caso não sejam válidos, o processo é encerrado e os mensagens erros são retornadas junto do status HTTP igual a 400, que indica erro na validação dos dados.

Por se tratar de uma Web API e se baseando nas boas práticas de programação segundo padrão SOLID (THELMA, 2020), o software foi desenvolvido de maneira segmentada com três partes principais, em que cada uma possui responsabilidades únicas. As partes e suas responsabilidades únicas são as seguintes: *DataContracts,* conter todos os contratos para as operações; *Core*, possuir todas as funcionalidades da API; *e Application*, cujas responsabilidades são ter registrado todos os serviços utilizados pelo código, conter as configurações necessárias para a API funcione corretamente e possuir os *controllers*, que são classes que permitem a integração de um determinado endpoint com alguma funcionalidade da API por meio do *framework* ASP.NET MVC. A parte *Core* da API também possui subdivisões que seguem a mesma premissa de responsabilidade única. Dessa forma, este foi dividido em pasta contendo cada funcionalidades da aplicação, sendo as principais descritas a seguir.

Além disso, testes de unidade para cada funcionalidade foram desenvolvidos com o intuito de validar as lógicas. Segundo (NOÇÕES BÁSICAS DE TESTE DE UNIDADE, 2019), o teste de unidade consiste em testes isolados de cada parte de uma determinada funcionalidade, com o objetivo de assegurar sua validação e que funciona conforme o esperado, além de servir para detectar possíveis erros, inconsistências e *bugs* no sistema.

**2.1 CÁLCULO DAS REAÇÕES DO SISTEMA DE SUSPENSÃO**

Para o cálculo das reações, o sistema de suspensão foi considerado como uma treliça com os amortecedores atuando como barras rígidas, como mostrado na Fig. 2, em que R representa as reações nos componentes e u, os vetores normalizados de cada componente. Assim, foram calculadas as equações de equilíbrio do corpo considerando uma força e um momento aplicada no ponto O, em que se obteve um sistema linear de 6 equações que foi transformado em uma operação com matrizes, conforme será apresentado a seguir.

Imagem de vídeo game

Descrição gerada automaticamente com confiança média

Figura 2 – Sistema de suspensão para cálculo das reações (SCHOMMER, 2021)

|  |  |
| --- | --- |
|  | (1) |

|  |  |
| --- | --- |
|  | (2) |

Separando estas equações nos eixos x, y e z, temos:

Transformando o sistema acima em uma equação matricial do tipo , em que [A] é uma matriz que representa a geometria de suspensão, [X] é a matriz de reações nos componentes e [B] é uma matriz de forças e momentos aplicados ao sistema, tem-se:

Com as matrizes formadas é possível manipular a equação matricial inicial para encontrar os valores das reações nos componentes de suspensão, realizando o processo de inversão da matriz deslocamento e multiplicando pela matriz de esforços aplicados. Formando a seguinte equação apresentada abaixo. Ademais, vale salientar que para o cálculo da inversa da matriz [A] foi utilizado o método de Gauss-Jordan.

|  |  |
| --- | --- |
|  | (3) |

Para a realização dessas etapas, foi desenvolvida a operação *CalculateReactions*, em que a Fig. 3 mostra o fluxograma desta.

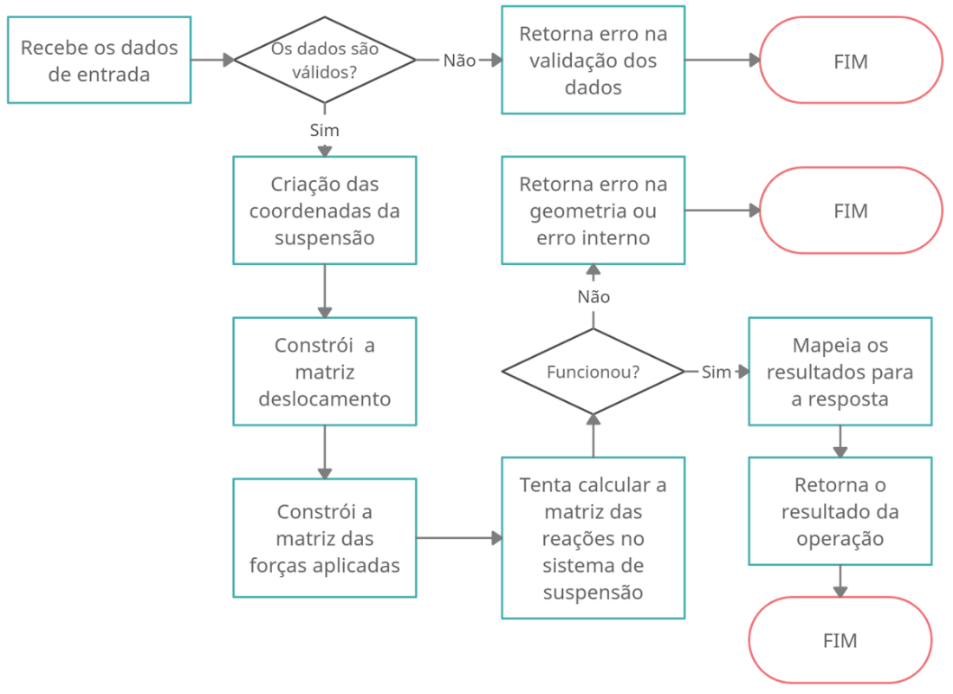


Figura 3 – Fluxograma da operação de cálculo das reações

Como apresentado na Fig. 3, no início da operação, a API recebe os dados de entrada que consistem nos pontos de suspensão e valor e ponto de aplicação da força e momento. Logo em seguida, são feitas validações de tipo em cada parâmetro de modo a assegurar que podem ser usados ao longo da operação. Sendo válidos, a operação continua a executar, retornando ao final os valores absoluto e em cada eixo das reações em cada componentes.

**2.2 EQUAÇÕES CONSTITUTIVAS DE RESISTÊNCIA DOS MATERIAIS**

Segundo (Hibbeler, 2009), a resistência dos materiais é um ramo da engenharia mecânica que estuda as relações entre esforço aplicado a um corpo deformável e a intensidade das forças internas atuantes no corpo. Esse tema compreende o cálculo da deformação de corpos e o estudo da sua estabilidade e tensões, quando ele é submetido a esforços externos. Portanto, para a análise estática do sistema de suspensão foram consideradas as equações constitutivas de resistência dos materiais, já que com elas é possível obter os valores de tensões que o componente estrutural sofrerá, considerando seu material, perfil, força aplicada e tipo de fixação. De acordo com (Hibbeler, 2009), foi extraído as equações para cálculo de tensão normal, tensão equivalente e força crítica de flambagem em cada componente do sistema de suspensão.

Seguindo as boas práticas e o padrão SOLID, foi implementada a classe *MechanicsOfMaterials* com a responsabilidade única de conter as fórmulas de resistência dos materiais que são usadas em outras funcionalidades do *software*, como a operação que realiza a análise estática no sistema de suspensão.

Como os componentes de suspensão que foram analisados possuem seção transversal constante, foi utilizado a equação mais simples para o cálculo de tensão normal, conforme demonstrado abaixo.

|  |  |
| --- | --- |
|  | (4) |

Onde:

σ - Tensão normal média em qualquer ponto de área da seção transversal.

F - Força normal interna resultante, que é aplicada no centroide da área da seção transversal.

A - Área da seção transversal da barra.

Para o cálculo da tensão equivalente, foi utilizada a equação de Von Misses, conforme apresentado abaixo.

|  |  |
| --- | --- |
|  | (5) |

Onde:

- Tensão fletora em x.

- Tensão fletora em y.

- Tensão de cisalhamento no plano XY.

Contudo, como neste estudo não são consideradas as tensões no eixo y, podendo reescrever a equação (5) conforme apresentado a seguir, em que representa a tensão fletora em um eixo e a tensão cisalhante em um plano.

|  |  |
| --- | --- |
|  | (6) |

Devido a consideração de treliça na suspensão, implementar o cálculo da força crítica de flambagem apresentou grande importância, sendo utilizada a equação apresentada abaixo.

|  |  |
| --- | --- |
|  | (7) |

Onde:

- Módulo de Young

- Momento de inércia

- Comprimento do tubo

- Valor correspondente ao tipo de fixação

**2.3 ANÁLISE ESTÁTICA DO SISTEMA DE SUSPENSÃO**

Com o intuito de fornecer ao usuário uma análise estrutural simples e de rápida solução, em que é considerada apenas o material e o perfil, foi implementada a operação *RunStaticAnalysis*, que recebe os mesmos dados de entrada da operação, além do material dos componentes e do perfil de cada. O fluxograma para esta operação pode ser visto na Fig. 4.

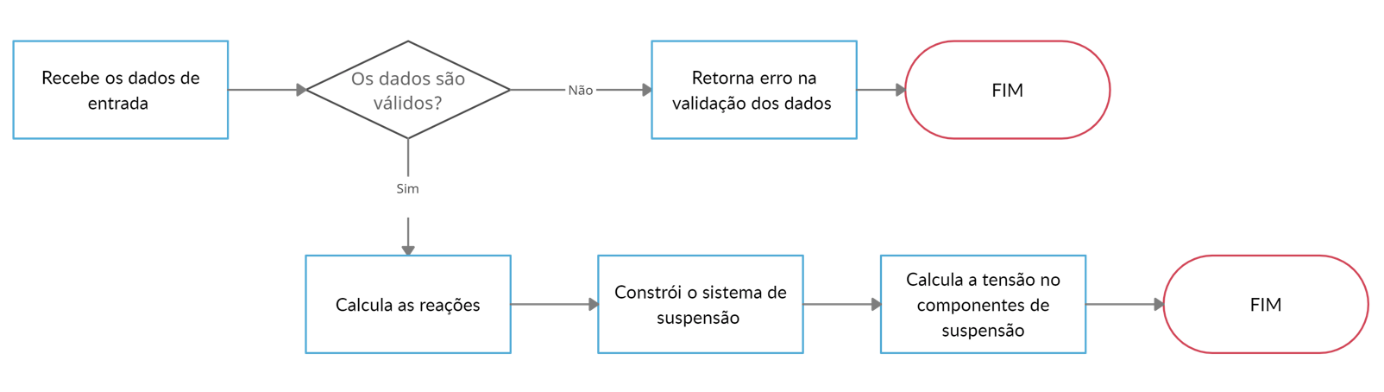


Figura 4 – Fluxograma da operação de análise estática

Conforme apresentado na Fig. 4, esta operação calcula as reações no sistema de suspensão chamando a operação *CalculateReactions*, já que não faz-se necessário reimplementar a lógica desta operação na classe *RunStaticAnalysis*. Depois disso, é construído um objeto que representa o sistema de suspensão para facilitar as manipulações internas para, por último, calcular as tensões em cada componente e o fator de segurança à flambagem. Vale salientar que os valores de tensão equivalente, fator de segurança, força crítica de flambagem e fator de segurança à flambagem de cada componente é retornado como dado de resposta.

**3 CONCLUSÕES E TRABALHOS FUTUROS**

Mediante ao apresentado, o *software* teve sua primeira versãodesenvolvida e lançada para analisar estruturalmente os componentes do veículo Baja SAE. A equipe Mud Runner Baja, que foi a equipe que utilizou a API desenvolvida, obteve resultados mais precisos em comparação aos cálculos analíticos que utilizavam anteriormente.

A aplicação foi desenvolvida para rodar localmente através da plataforma *Swagger*, como pode ser visto nas Figuras 5 e 6,que facilita o uso do software para os clientes, já que não são necessários conhecimentos de programação para utilizar a aplicação nessa plataforma. No *Swagger*,o cliente passa os dados de entrada definidos nos contratos do programa e recebe os resultados da operação em poucos segundos, aumentando, portanto, além da precisão dos cálculos, a velocidade do processo de análise.

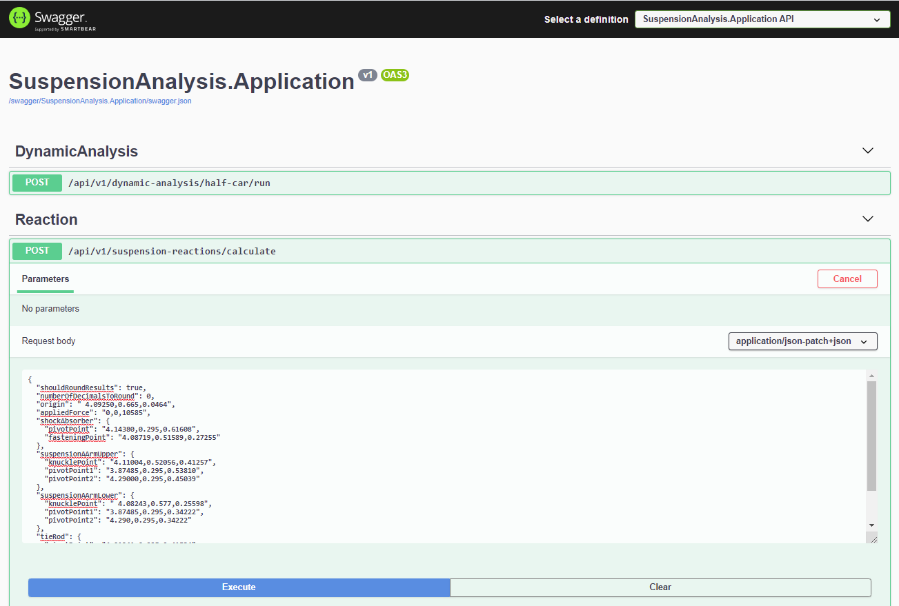


Figura 5 – Entrada da aplicação passada no *Swagger*

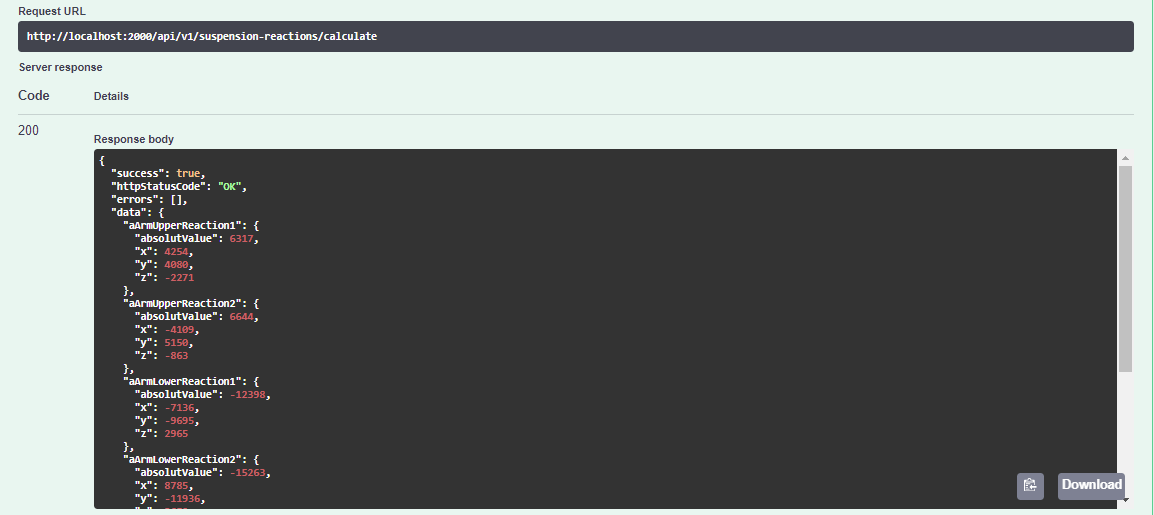


Figura 6 – Resultados da aplicação retornados no *Swagger*

A Fig. 5 representa o *Swagger* recebendo as entradas da operação de cálculo das reações. Já a Fig. 6 representa os resultados da aplicação.

Quanto a operação de análise estática do sistema de suspensão, a equipe encontrou valores de tensões menores que as tensões encontradas nas análises de elemento finitos. Contudo, mesmo assim os resultados foram considerados coerentes, já que no *software* desenvolvido não são levados em consideração quaisquer tipos de concentradores de tensão, logo as tensões retornadas como resultados serão mais baixas que as reais e que as encontradas em análises de elementos finitos. O objetivo dessa análise realizada na aplicação é de visualizar em uma análise prévia a análise de elementos finitos se o componente pode falhar com os esforços recebidos. Caso, a geometria da peça falhe nessa etapa de análise, a geometria nem passa para uma análise de elementos finitos, economizando tempo e esforço da equipe que fez uso da API. Portanto, o software desenvolvido e apresentado nesse estudo, teve sua primeira versão utilizada pela Equipe Mud Runner Baja e todas as funcionalidades propostas em sua primeira versão foram utilizadas e apresentaram resultados satisfatórios para o cliente.

Os próximos passos do presente estudo visam aumentar a aplicabilidade do *software*. O primeiro deles é calcular os esforços presentes na manga de eixo e no cubo de roda do protótipo Baja, para isso os estudos da manga já se encontram em fase de revisão, já para o cubo ainda é necessário o desenvolvimento de toda lógica de programação.

Em seguida as atenções são direcionadas para a análise de dinâmica vibracional para meio-carro com 4, 6 e 8 graus de liberdade. Para 4 graus de liberdade faltam algumas implementações na lógica de programação, mas há algumas equações finalizadas. Para 6 graus de liberdade o estudo se encontra bem evoluído, restando apenas a revisão e teste de unidade. Para 8 graus de liberdade todo o desenvolvimento ainda precisa ser feito. Por fim, para uma análise vibracional mais completa, pretende-se considerar o carro por completo, envolvendo vários graus de liberdade, em que a quantidade exata ainda precisa ser decidida em novos estudos para compreender o que de fato pode ser vantajoso para uma equipe de Baja SAE. Estas funcionalidades são de suma importância para um melhor dimensionamento do sistema de suspensão, visto que será considerado a atuação do amortecedor.

Pretende-se também a implementação das rotinas de outros subsistemas para que a Web API apresentada neste estudo possa ser usada como ferramenta em todo o desenvolvimento do projeto de um Baja SAE. Vale salientar que as rotinas para o projeto do sistema de freio e trem de força já foram mapeadas, faltando somente a definição dos contratos e fronteiras de cada funcionalidade para que a implementação possa ser concluída.

Além disso, a integração do programa desenvolvido com outros *softwares* de análises é de grande interesse. Já que integrando a API com o Ansys APDL, que é mais leve e rápido que o Ansys Workbench, os dados obtidos através da aplicação podem ser compartilhados de uma forma automatizada para o Ansys APDL, que forneceria uma análise prévia melhor para a avaliação numérica.

Visando ainda, uma melhoria nos cálculos estruturais, uma integração com o Ansys Workbench e o Adams Car é objetivada para evitar trabalhos repetitivos nas análises. Contudo, ainda é necessário compreender se a complexidade e o esforço de automação desse processo são compensados pelos benefícios, como principal a redução de trabalho repetitivo que forneceria. Caso a relação anterior seja de maior benefício que esforço realizado, essa integração será o último passo de incremento, planejado até o momento da publicação desse artigo.

**REFERÊNCIAS BIBLIOGRÁFICAS**

DE CAMARGO, Wellington Balbo. **Conceitos e Exemplos – Herança: Programação Orientada a Objetos – Parte 1.** DEVMEDIA. 2010**.**Disponível em:https://www.devmedia.com.br/conceitos-e-exemplos-heranca-programacao-orientada-a-objetos-parte-1/18579. Acessado em 23 maio 2021.

DRUNN, J. D. **Estudo e desenvolvimento de um sistema de transmissão automatizada para um veículo de competição baja sae.** 2018. Tese de conclusão de curso em Engenharia Mecânica. Universidade Regional do Noroeste do Estado do Rio Grande do Sul - UNIJUÍ. PANAMBI. Disponível em:

<https://bibliodigital.unijui.edu.br:8443/xmlui/bitstream/handle/123456789/5911/Jonathan%20Drunn.pdf?sequence=1&isAllowed=y>. Acessado em: 5 jul. 2021.

FERNADES, M.V.D.S.F. **Análises da suspensão e direção de um veículo "off-road" do tipo mini baja.** 2015. Monografia do Projeto de Pesquisa apresentada à disciplina de Trabalho de Conclusão de Curso –TCC 2 do curso de Engenharia Mecânica. Universidade Tecnológica Federal do Paraná. UTFPR. Disponível em: http://riut.utfpr.edu.br/jspui/handle/1/10268. Acessado em: 07 jul. 2021.

FERREIRA, E.F. **Influências do projeto baja sae no ensino da engeharia e no desenvolvimento do aluno.** 2011. Trabalho de Graduação em Engenharia Mecânica. Universidade Estadual Paulista. UNESP. Disponível em: https://repositorio.unesp.br/bitstream/handle/11449/119038/ferreira\_eg\_tcc\_guara.pdf?sequence=1&isAllowed=y. Acessado em: 05 jul. 2021.

GASPAROTTO, H. M. **Introdução ao ASP.NET MVC.**DEVMEDIA. 2014. Disponível em: devmedia.com.br/introducao-ao-asp-net-mvc/31878. Acessado em: 29 abr. 2021

MICROSOFT.**APIS +MICROSERVICES: The ultimate guide to creating an enterprise API platform.** 2019. Disponível em: https://azure.microsoft.com/mediahandler/files/resourcefiles/apis-microservices-ebook/Azure\_API-Microservices\_eBook.pdf. Acessado em: 27 abr. 2021.

MICROSOFT. **Um tour pela linguagem C#.** 2021. Disponível em: https://docs.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/. Acessado em: 27 abr. 2021.

MINOZZO, R.M. **Aquisição de forças de dinâmicas atuantes na suspensão dianteira de um veículo protótipo baja sae.** 2012. Trabalho de Graduação em Engenharia Mecânica. Universidade Federal do Rio Grande do Sul. UFRGS. Disponível em:

https://www.lume.ufrgs.br/handle/10183/61988. Acessado em: 06 jul. 2021.

MIRLISENNA, G. **Método dos elementos finitos: o que é?** 2016. ESSS. Disponível em: https://www.esss.co/blog/metodo-dos-elementos-finitos-o-que-e/. Acessado em: 13 jul. 2021

NAVARRO, F. **Introdução ao ASP.NET Web API.** 2012. Disponível em: https://www.devmedia.com.br/introducao-ao-asp-net-web-api/25180. Acessado em 5 jul. 2021.

NOÇÕES BÁSICAS DE TESTE DE UNIDADE. 2019. Microsoft. Disponível em: https://docs.microsoft.com/pt-br/visualstudio/test/unit-test-basics?view=vs-2019. Acessado em: 14 jul. 2021.

REIS, F. A.; BUSTAMANTE S.B.V..**MINI BAJA: Automação e Desenvolvimento de Artefatos.** 2003. Universidade Católica de Petrópolis. Disponível em: http://br-ie.org/pub/index.php/wie/article/view/796/782. Acessado em: 05 jul. 2021.

RICK ANDERSON. **Visão geral do ASP.NET.** Microsoft. 2019. Disponível em: https://docs.microsoft.com/pt-br/aspnet/overview. Acessado em: 29 abr. 2021.

SAE BRASIL. **Baja Nacional.** 2021. Disponível em: https://saebrasil.org.br/programas-estudantis/baja-sae-brasil/. Acessado em: 30 jun. 2021.

SCHOMMER, A. **Static Equilibrium of a Double Wishbone Suspension.** 2021. GitHub. Disponível em: https://github.com/adrianoschommer/static\_equilibrium\_suspension. Acessado em: 06 jul. 2021.

SENA, P.C.P. **Automação de processos de projeto de programação em BIM: Dynamo, Pyhon e C#.** 2019. Dissertação para obtenção do título de mestre em Arquitetura e Urbanismo. Universidade de São Paulo. Disponível em: https://www.teses.usp.br/teses/disponiveis/102/102131/tde-12032020-144132/publico/DissCorrigidaPauloCezarPeixotoSena.pdf . Acessado em: 07 jul. 2021

SMARTBEAR. **About Swagger.** 2021. Disponível em: https://swagger.io/about/. Acessado em: 29 abr. 2021.

TASK CLASSE. 2021. Microsoft. Disponível em: https://docs.microsoft.com/pt-br/dotnet/api/system.threading.tasks.task?view=net-5.0. Acessado em: 4 maio 2021.

TECHTUDO. **O que é API e para que serve? Cinco perguntas e respostas**. 2020. Disponível em: https://www.techtudo.com.br/listas/2020/06/o-que-e-api-e-para-que-serve-cinco-perguntas-e-respostas.ghtml. Acessado em: 27 abr. 2021.

THELMA, Ugonna. **The S.O.L.I.D Principles in Pictures.** 2020. Disponível em: https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898. Acessado em 05 jul. 2021.

**Dados dos autores:**

**Bruno Mello Silveira**. Graduando em Engenharia Mecânica. Centro Federal de Educação Tecnológica Celso Suckow da Fonseca – CEFET/RJ. Estudante. Rua General Canabarro, 485 - Maracanã, Rio de Janeiro – RJ. 21970000933. bruno.silveira@aluno.cefet-rj.br.

**Emerson Marins Marques Souvain**. Graduando em Engenharia Mecânica. Centro Federal de Educação Tecnológica Celso Suckow da Fonseca – CEFET/RJ. Estudante. Rua General Canabarro, 485 - Maracanã, Rio de Janeiro – RJ. 21981000572. emerson.souvain@aluno.cefet-rj.br.

**Marcus Vinícius Mariath Machado Nogueira**. Graduando em Engenharia de Controle e Automação. Centro Federal de Educação Tecnológica Celso Suckow da Fonseca – CEFET/RJ. Estudante. Rua General Canabarro, 485 - Maracanã, Rio de Janeiro – RJ. 21976739569. marcus.nogueira@aluno.cefet-rj.br.

Indicação da seção em que o(s) autor(es) deseja(m) ter seu texto publicado: ( )T&S ( ) T&G (X) T&I