

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
GOIANO – CAMPUS RIO VERDE - GO
PROGRAMA DE PÓS-GRADUAÇÃO *STRICTO SENSU* EM
ENGENHARIA APLICADA E SUSTENTABILIDADE**

**UMA PROVA DE CONCEITO DE UM APLICATIVO WEB PARA
ANÁLISE DE PÓRTICOS PLANOS**

Autor: Charles França de Sousa
Orientador: Prof. Dr. Marcel Willian Reis Sales
Coorientador: Prof. Dr. Michell Macedo Alves

RIO VERDE - GO
Agosto - 2023

CHARLES FRANÇA DE SOUSA

**UMA PROVA DE CONCEITO DE UM APLICATIVO WEB PARA
ANÁLISE DE PÓRTICOS PLANOS**

Projeto de qualificação apresentado, como parte das exigências para obtenção do título de MESTRE EM ENGENHARIA APLICADA E SUSTENTABILIDADE, no Programa de Pós-Graduação em Engenharia Aplicada e Sustentabilidade do Instituto Federal de Educação, Ciência e Tecnologia Goiano – Campos Rio Verde – Área de concentração (Tecnologia e Ciência dos Materiais).

Autor: Charles França de Sousa

Orientador: Prof. Dr. Marcel Willian Reis Sales

Coorientador: Prof. Dr. Michell Macedo Alves

RIO VERDE - GO

Agosto - 2023

Sistema desenvolvido pelo ICMC/USP
Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas - Instituto Federal Goiano

SC477p Sousa, Charles França de
UMA PROVA DE CONCEITO DE UM APLICATIVO WEB PARA
ANÁLISE DE PÓRTICOS PLANOS / Charles França de Sousa;
orientador Marcel Willian Reis Sales; co-orientador
Michell Macedo Alves. -- Rio Verde, 2023.
103 p.

Dissertação (Mestrado em Mestrado Profissional em
Engenharia Aplicada e Sustentabilidade) -- Instituto
Federal Goiano, Campus Rio Verde, 2023.

1. elementos pórticos. 2. ensino-aprendizagem. 3.
orientação a objeto. 4. software educacional. 5.
geração de malhas. I. Willian Reis Sales, Marcel,
orient. II. Macedo Alves, Michell, co-orient. III.
Título.

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO

PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS

NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO

Com base no disposto na Lei Federal nº 9.610, de 19 de fevereiro de 1998, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano a disponibilizar gratuitamente o documento em formato digital no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

IDENTIFICAÇÃO DA PRODUÇÃO TÉCNICO-CIENTÍFICA

- | | |
|--|---|
| <input type="checkbox"/> Tese (doutorado) | <input type="checkbox"/> Artigo científico |
| <input checked="" type="checkbox"/> Dissertação (mestrado) | <input type="checkbox"/> Capítulo de livro |
| <input type="checkbox"/> Monografia (especialização) | <input type="checkbox"/> Livro |
| <input type="checkbox"/> TCC (graduação) | <input type="checkbox"/> Trabalho apresentado em evento |

Produto técnico e educacional - Tipo:

Nome completo do autor:

Charles França de Sousa

Matrícula:

2021102331440060

Título do trabalho:

UMA PROVA DE CONCEITO DE UM APLICATIVO WEB PARA
ANÁLISE DE PÓRTICOS PLANOS

RESTRIÇÕES DE ACESSO AO DOCUMENTO

Documento confidencial: Não Sim, justifique:

Informe a data que poderá ser disponibilizado no RIIF Goiano: 22 / 12 / 2023

O documento está sujeito a registro de patente? Sim Não

O documento pode vir a ser publicado como livro? Sim Não

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O(a) referido(a) autor(a) declara:

- Que o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- Que obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autoria, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- Que cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Rio Verde - GO

15 / 12 / 2023

Local

Data



Documento assinado digitalmente
CHARLES FRANCA DE SOUSA
Data: 30/01/2024 10:24:03-0300
Verifique em <https://validar.iti.gov.br>

Assinatura do autor e/ou detentor dos direitos autorais

Ciente e de acordo:


Assinatura do(a) orientador(a)



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

Ata nº 55/2023 - GGRAD-RV/DE-RV/CMPRV/IFGOIANO

PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
ATA DE DEFESA - PÓS-GRADUAÇÃO

Unidade do IF Goiano:	Campus Rio Verde	
Programa de Pós-Graduação:	Engenharia Aplicada e Sustentabilidade	
Defesa de:	Dissertação	Defesa de número: 66
Data: 01/09/2023	Hora de início: 09:00h	Hora de encerramento: 12:00h
Matrícula do discente:	2021102331440060	
Nome do discente:	Charles França de Sousa	
	UMA PROVA DE CONCEITO DE UM APLICATIVO WEB PARA ANÁLISE DE	

Título do trabalho:	PÓRTICOS PLANOS USANDO JAVASCRIPT
Orientador:	Marcel Willian Reis Sales
Área de concentração:	Engenharia Aplicada e Sustentabilidade
Linha de Pesquisa:	Tecnologia e Ciência dos Materiais
Projeto de pesquisa de vinculação	Aplicação web com interface gráfica para ensino de métodos dos elementos finitos usando javascript
Titulação:	Mestre em Engenharia Aplicada e Sustentabilidade

Nesta data, reuniram-se os componentes da Banca Examinadora, Prof. Dr. Marcel Willian Reis Sales (Presidente da banca), Prof. Dr. Wanderlei Malaquias Pereira Junior (Avaliador externo), Prof. Dr. Luiz Fernando Martha (Avaliador Externo) Prof. Dr. Michell Macedo Alves (Avaliador Interno) sob a presidência do primeiro, em sessão pública realizada de forma presencial, para procederem a avaliação da defesa de dissertação, em nível de Mestrado, de autoria de **CHARLES FRANÇA DE SOUSA**, discente do Programa de Pós-Graduação em Engenharia Aplicada e Sustentabilidade do Instituto Federal Goiano – Campus Rio Verde. A sessão foi aberta pelo presidente da Banca Examinadora, Marcel Willian Reis Sales, que fez a apresentação formal dos membros da Banca. A palavra, a seguir, foi concedida o(a) autor (a) da dissertação para, em 40 min., proceder à apresentação de seu trabalho. Terminada a apresentação, cada membro da banca arguiu o(a) examinado(a), tendo-se adotado o sistema de diálogo sequencial. Terminada a fase de arguição, procedeu-se a avaliação da defesa. Tendo-se em vista as normas que regulamentam o Programa de Pós-Graduação em Engenharia Aplicada e Sustentabilidade, e procedidas às correções recomendadas, a dissertação foi APROVADA, considerando-se integralmente cumprido este requisito para fins de obtenção do título de **MESTRE EM ENGENHARIA APLICADA E SUSTENTABILIDADE**. A conclusão do curso dar-se-á quando da entrega na secretaria do PPGEAS da versão definitiva da dissertação, com as devidas correções. Assim sendo, esta ata perderá a validade se não cumprida essa condição, em até **60** (sessenta) dias da sua ocorrência. A Banca Examinadora recomendou a publicação dos artigos científicos oriundos dessa Tese em periódicos de circulação nacional e/ou internacional, após procedida as modificações sugeridas. Cumpridas as formalidades da pauta, a presidência da mesa encerrou esta sessão de defesa de dissertação de mestrado, e foi lavrada a presente Ata, que, após lida e achada conforme, será assinada pelos membros da Banca Examinadora.

Decisão da banca: Aprovada

Esta defesa é parte dos requisitos necessários para a obtenção do título de Mestre em Engenharia Aplicada e Sustentabilidade.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna

do IFGoiano.

Documento assinado eletronicamente por:

- Marcel Willian Reis Sales, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 07/11/2023 16:25:19.
- Luiz Fernando Martha, Luiz Fernando Martha - 234310 - Professor de engenharia - Instituto Federal Goiano - Campus Rio Verde (10651417000500), em 07/11/2023 17:52:02.
- Wanderlei Malaquias Pereira Junior, Wanderlei Malaquias Pereira Junior - Professor Avaliador de Banca - Universidade Federal de Catalão (35834377000120), em 07/11/2023 18:53:59.
- Michell Macedo Alves, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 07/11/2023 18:59:19.

Este documento foi emitido pelo SUAP em 07/11/2023. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 545426

Código de Autenticação: fe5f86b360



INSTITUTO FEDERAL GOIANO

Campus Rio Verde

Rodovia Sul Goiana, Km 01, Zona Rural, 01, Zona Rural, RIO VERDE / GO, CEP 75901-970

(64) 3624-1000



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

Documentos 83/2023 - SREPG/CMPR/CPG-RV/DPGPI-RV/CMPRV/IFGOIANO

UMA PROVA DE CONCEITO DE UM APLICATIVO WEB PARA ANÁLISE DE PÓRTICOS PLANOS USANDO JAVASCRIPT.

Autor: Charles França de Sousa
Orientador: Prof. Dr. Marcel Willian Reis Sales

TITULAÇÃO: Mestre em Engenharia Aplicada e Sustentabilidade - Área de Concentração Engenharia Aplicada e Sustentabilidade

APROVADO em 01 de setembro de 2023.

Prof. Dr. Wanderlei Malaquias Pereira
Junior
Avaliador Externo - Universidade
Federal de Catalão – UFCAT

Prof. Dr. Luiz Fernando Martha
Avaliador Externo - Instituto
Tecgraf (PUC-Rio)

Prof. Dr. Michell Macedo Alves
Avaliador Interno - IFGOIANO / Rio
Verde

Prof. Dr. Marcel Willian Reis Sales
Presidente da banca - IFGOIANO / Rio
Verde

Documento assinado eletronicamente por:

- Luiz Fernando Martha, Luiz Fernando Martha - 234310 - Professor de engenharia - Instituto Federal Goiano - Campus Rio Verde (10651417000500), em 16/02/2024 10:53:18.
- Marcel Willian Reis Sales, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 06/11/2023 18:00:34.
- Michell Macedo Alves, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 06/11/2023 18:05:22.
- Wanderlei Malaquias Pereira Junior, Wanderlei Malaquias Pereira Junior - Professor Avaliador de Banca - Universidade Federal de Catalão (35834377000120), em 07/11/2023 11:05:33.

Este documento foi emitido pelo SUAP em 31/08/2023. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 527274
Código de Autenticação: eb7378ee4f



INSTITUTO FEDERAL GOIANO
Campus Rio Verde
Rodovia Sul Goiana, Km 01, Zona Rural, 01, Zona Rural, RIO VERDE / GO, CEP 75901-970
(64) 3624-1000

DEDICATÓRIA

Aos meus pais, Joabi e Shirleys

À minha irmã, Ister

À minha esposa, Jusciely

Ao meu avô, Plinio (In memoria)

AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter me concedido saúde e livre arbítrio para continuar essa caminhada.

Aos meus pais, pelo incentivo e apoio incondicional. Agradeço a minha mãe, Shirleys Silva França uma heroína que me concedeu total apoio e incentivo, mesmo não estando entre nós sempre será uma heroína. Ao meu pai, Joabi Ribeiro da Sousa, apesar das dificuldades me fortaleceu ao passo que se tornou essencial em minha vida para que eu chegasse aos meus objetivos.

Gratidão a minha segunda mãe do coração, Vanusia Carneiro pelo carinho e pelo apoio que sempre recebi nos últimos sete anos, que sempre irei lembrar e agradecer por tudo. Agradecer minha esposa Jusciely em meio a dificuldades sempre esteve no meu lado dando apoio e com muito carinho.

Aos meus professores, orientador e coorientador, Marcel Willian Reis Sales e Michell Macedo Alves, que sempre estiveram dispostos a ajudar em todos os momentos, sempre com bons conselhos e muita paciência. Obrigado por contribuir para me tornar um pesquisador e um ser melhor.

Aos meus colegas do PPGEAS Fernando, Bruno, Matheus, obrigado pelo companheirismo em meio as dificuldades que o mestrado apresentou. Aos meus amigos da graduação e professor que sempre esteve ao meu lado Alan Gomes, Elielton Santana, Rogerio Pereira.

Ao IFGoiano que me acolheu no Campus - Rio Verde. A todos os docentes do PPGEAS onde só tenho a agradecer.

RESUMO

O método dos elementos finitos tem sido usado para diversos fins na engenharia, mas principalmente para análise estrutural. Neste sentido, o ensino deste método tem evoluído para cada vez mais auxiliar no aprendizado dos estudantes. Por isso, o uso de softwares em sala de aula tem se tornado uma excelente forma de ensinar e aplicar modelos práticos na engenharia. Apesar disso, a maioria dos softwares para refinamento de malhas, são aplicações para desktops, e são de alto custo financeiro. Desta maneira, este trabalho tem como objetivo a propor uma aplicação web com computação gráfica com a finalidade de ser usado de forma educacional para ensino e análise pórticos planos. Para o desenvolvimento da revisão sistemática foi utilizado o software StArt, possuindo três etapas, sendo elas: 1) o protocolo que é o planejamento do estudo, execução que é responsável pela identificação, 2) seleção e extração dos estudos e 3) sumarização, que é composto pelos gráficos e resultado da revisão. Os resultados mostram que há uma quantidade diversificada de softwares web para auxílio do ensino em diversas áreas, entretanto para o ensino de métodos de elementos finitos com engenharias de estruturas em web, que seja gratuito, não foi encontrado ocasionando uma carência dessa aplicação. No capítulo 02 desta dissertação está previsto para ser concluso o desenvolvimento do software educacional WEB, para auxílio dos alunos de graduação e pós-graduação no desenvolvimento de método dos elementos finitos de forma online e gratuita.

Palavras-chave: software educacional; orientação a objeto; geração de malhas; ensino-aprendizagem; elementos pórticos.

ABSTRACT

The finite element method has been used for various purposes in engineering, but mainly for structural analysis. In this sense, the teaching of this method has evolved to increasingly assist students in learning. Therefore, the use of software in the classroom has become an excellent way to teach and apply practical engineering models. Despite this, most software for mesh refinement are desktop applications and have a high financial cost. Therefore, this work aims to propose a web application with computer graphics with the purpose of being used in an educational way for teaching and analyzing flat frames. For the development of the systematic review, the StArt software was used, having three stages, namely: 1) the protocol which is the study planning, execution which is responsible for identification, 2) selection and selection of studies and 3) summarization, which It consists of graphics and review results. The results show that there is a distributed amount of web software to aid teaching in different areas, however for teaching finite element methods with structural engineering on the web, which, as it is free, was not found, causing a lack of this application. In chapter 02 of this dissertation, the development of the WEB educational software is expected to be completed, to assist undergraduate and postgraduate students in developing the finite element method online and free of charge.

Keywords: educational software; Object-oriented; mesh generation; teaching-learning; portico elements.

SUMÁRIO

LISTA DE FIGURAS	9
DE SIGLAS	10
1. INTRODUÇÃO GERAL	11
2. JUSTIFICATIVA	12
3. PROBLEMA	13
4. HIPÓTESES	13
5. OBJETIVOS	13
5.1. OBJETIVOS ESPECIFICOS	13
6. REVISÃO DE LITERATURA	14
6.1. SOFTWARES EDUCACIONAIS	14
6.2. SOFTWARES NA ENGENHARIA	15
6.3. APLICAÇÕES <i>WEB</i>	17
6.3.1. ARQUITETURA DO TCP/IP	19
6.3.2. APLICAÇÃO	20
6.3.4. INTERNET	22
6.3.5. ACESSO À REDE	22
7. METODOLOGIA	25
REFERÊNCIAS BIBLIOGRÁFICAS	27
CAPÍTULO 1 – ARTIGO 1	30
1 INTRODUÇÃO	33
1.2 MÉTODO	34
1.3 RESULTADOS E DISCUSSÕES	38
1.4 CONCLUSÃO/CONSIDERAÇÕES FINAIS	44

REFERÊNCIAS	44
CAPÍTULO 2 – ARTIGO 2	48
1 INTRODUÇÃO.....	51
2 REFERENCIAL TEÓRICO.....	52
2.1 SOFTWARE EDUCACIONAL.....	52
2.2 SOFTWARE EDUCACIONAL PARA ANÁLISES DE ESTRUTURAS.....	53
2.3 PROGRAMAÇÃO ORIENTADA A OBJETOS (POO).....	55
2.4 METODOS DOS ELEMENTOS FINITOS.....	55
3 MÉTODO	57
4 RESULTADOS E DISCUSSÕES.....	63
4.1 PÁGINA INICIAL DO SISTEMA MEFWEB.....	63
4.2 INTERFACE DO SISTEMA MEFWEB.....	65
5 CONCLUSÕES	70
REFERÊNCIAS	71
CONSIDERAÇÕES FINAIS	72
APÊNDICES E ANEXOS	74
APÊNDICE A - LISTA DE TRABALHOS DA EXTRAÇÃO DA RSL	74

LISTA DE FIGURAS

FIGURA 01 - Aplicação da equação vetorial do plano.....	15
FIGURA 02 - Pré-Processador, Processador e Pós-Processador.....	16
FIGURA 03 - Arquitetura cliente/servidor.....	18
FIGURA 04 - Interface de programação da aplicação socket.....	19
FIGURA 05 - Aplicação TCP/IP e seus protocolos.....	20
FIGURA 06 - Exemplo de Rede Ethernet.....	23
FIGURA 07 - Linguagens de programação mais utilizadas em 2021.....	24
FIGURA 08 – Padrão MVC.....	26
FIGURA 09 - Layout da aplicação.....	27

CAPITULO 1:

FIGURA 01 - Formulários de extração e qualidade da RSL.....	38
FIGURA 02 – Bases de dados.....	49
FIGURA 03 – Fluxograma das etapas da RSL.....	40
FIGURA 04 – Gráfico dos dados da extração.....	41

CAPITULO 2:

FIGURA 01 - Padrão MVC.....	58
FIGURA 02 – Diagrama de caso de uso.....	59
FIGURA 03 – Diagrama de classes.....	60
FIGURA 04 – Pagina inicial.....	64
FIGURA 05 – Pagina sobre.....	64
FIGURA 06 – Vantagens do MEFWEB.....	64
FIGURA 07 – Argumentos e contatos.....	65
FIGURA 08 – Interface do sistema.....	66
FIGURA 09 – Tipo do material.....	67
FIGURA 10 – Fluxograma MEFWEB.....	68
FIGURA 11 – Estrutura no MEFWEB.....	69

DE SIGLAS

APIs	– <i>Application Programming Interfaces</i>
CG	– Computação Gráfica
HTTP	– <i>Hypertext Transfer Protocol</i>
JS	– <i>JavaScript</i>
LaPES	– Laboratório de Pesquisa em Engenharia de <i>Software</i>
MEF	– Métodos dos Elementos Finitos
NASA	– Administração Nacional da Aeronáutica e Espaço
POO	– Programação Orientada a Objetos
PPC	– Projeto de Pesquisa Colaborativa
RSL	– Revisão Sistemática da Literatura
RV	– Realidade Virtual
StArt	– <i>State of the art through systematic review</i>
TDIC's	– Tecnologias Digitais da Informação e Comunicação
UFSCar	– Universidade Federal de São Carlos
URL	– <i>Uniform Resource Locator</i>
WEB	– <i>World Wide Web</i>
WebGL	– <i>Web Graphics Library</i>
IDE	– Ambiente de Desenvolvimento Integrado
UFSCar	– Universidade Federal de São Carlos
VSCoDe	– Visual Studio Code
MVC	– <i>Model-View-Controller</i>

1. INTRODUÇÃO GERAL

Há diversos tipos de ferramentas que podem auxiliar o professor e aluno na sala de aula desde do ensino da educação básica até o nível superior e pós graduação, e o uso das ferramentas digitais online é uma delas, possibilitando o ensino-aprendizado acessível a todos os alunos em todos os modelos de ensino desde o EAD, híbrido e presencial. Vale evidenciar que para o ensino de método dos elementos finitos nos cursos de engenharias na graduação e pós-graduação é indispensável o uso de *softwares*.

É notório que os alunos encontram diversas dificuldades em algumas disciplinas desde da graduação até na pós-graduação, entretanto em um mercado de trabalho cada vez mais competitivo se torna imprescindível o uso de *softwares* educacional para formação de profissionais.

O método dos elementos finito (MEF) teve seu surgimento nos meado do século XX, no ano de 1950 ele era utilizado para soluções de problemas à mecânica dos sólidos. No entanto, se popularizou em 1960, cinco anos após a NASA observar que seria de suma importância o desenvolvimento de um *software* para cálculo em estruturas gerais. A partir deste momento houve imensa propagação em diversas áreas de estudos como, engenharias, medicina, odontologia e em automóveis de competição.

O método dos elementos finitos, corresponde a dividir uma determinada estrutura em pequenas parcelas chamadas de elemento. Onde suas malhas estão interligadas pelos nós ou pontos nodais, este nós encontram-se sempre localizados nas vértices e lados do elemento, podendo variar de acordo com elemento finito. Desta maneira é feito os desenhos das estruturas na interface dos softwares, devem selecionar o método finito onde os pontos nodais devem ser bem definidos para que não aconteça erros no resultado da malha. Essas malhas podem ser poliedros, triângulos e tetraédrico, depende da estrutura que vai ser refinada para geração dos cálculos e malhas do MEF, MOREIRA (2020).

Os softwares que geram o refinamento de malhas consistem em pré-processador, processador e pós-processador. Estes softwares tem sido desenvolvidos para aplicação desktops com intuito para que o processamento dos dados do refinamento das malhas e das soluções

numéricas sejam mais rápidos. No caso de refinamento de malhas quanto maior os números de ponto nodais maior será o custo computacional do processamento.

Tendo em vista que atualmente é necessário agilidade e dinamismo para os resultados, o uso da aplicação web onde o usuário pode utilizar no escritório, e na sala de aula flexibilizando o conteúdo aplicado.

Dentro deste panorama, o presente trabalho apresenta o desenvolvimento de uma ferramenta educacional web com uma coleção de funções, tal como geração de pontos nodais, desenhos de figuras geométricas, criação de elementos pórticos e desenvolvimentos de barras. Esta aplicação web vai estar disponível para que alunos da matéria de elementos finitos possam aprender de forma prática com exemplos aplicados.

2. JUSTIFICATIVA

As tecnologias vêm proporcionando uma revolução em todas as áreas de conhecimentos desde o final do século XX. Nos cursos de graduação e pós-graduação não é diferente essa revolução, sendo um dos campos que mais avançou nos últimos anos em inovação do ensino e aprendizagem, sucedendo por meios de gamificação das aulas, softwares educacionais e softwares interativos, para proporcionar a interação, engajamento e auxílio para soluções em determinadas atividades.

Portanto, para o ensino nos cursos de graduação e pós-graduação de engenharia, existem certas disciplinas que os discentes apresentam maiores dificuldades, sendo matemática aplicada, análise estrutural entre outras. Contudo com auxílio de softwares educacionais ou de simulação essas disciplinas se tornam mais interativa com as possibilidades de pesquisa e simulações físicas. Entretanto, as tecnologias digitais não vêm para diminuir o trabalho do professor e sim, para auxiliar o aluno no ensino de aprendizagem quebrando o paradigma do ensino tradicional.

Com o desenvolvimento de diferentes softwares para diversas áreas de conhecimento, esta aplicação web para MEF tem intuito de auxiliar o professor na transmissão do conhecimento. Porém poucos alunos tem acesso a um computador com alta capacidade de processamento para utilizar determinada ferramenta, além disso, dispor de um software comercial de alto custo comercial não é a realidade da maioria das universidades.

Sob esta perspectiva, o presente projeto propõe o desenvolvimento de uma prova de conceito de um aplicativo web para análise de pórticos planos, com intuito de software

educacional para ensino nos cursos de métodos dos elementos finitos, nos de graduação e pós-graduação de engenharia civil, que seja gratuito e online com uma interface de fácil uso, amigável e possibilitando versões atualizada com novas funções.

3. PROBLEMA

Sabe-se que os métodos dos elementos finitos (MEF), contribuem diretamente para a comunidade com refinamento em engenharia de estruturas, análises estruturais do fêmur, modelagem de estrutura óssea, porém com a necessidade de um computador com softwares específicos para refinamento em MEF. Partindo deste pressuposto, o presente trabalho busca responder a seguinte pergunta de pesquisa: “Se seria possível implementar computação gráfica aplicada aos métodos de elementos finitos para plataforma web aplicada a engenharia de estruturas com fins educacionais?”.

4. HIPÓTESES

- Uma aplicação web para fins educacionais não tem alto custo de processamento, e pode ser usado para o ensino em sala de aula e trabalhos acadêmicos.
- A aplicação web pode ser rápida o suficiente para ser usada no navegador, que historicamente, é considerado lento para operações práticas em método dos elementos finitos.
- O desenvolvimento de uma plataforma web educacional para geração de malhas do MEF torna o ensino mais acessível, e irá ajudar os docentes dos cursos de engenharia na otimização dos seus estudos.

5. OBJETIVOS

- Desenvolver uma ferramenta educacional web orientada a objeto para análise e desenvolvimento de MEF;

5.1. OBJETIVOS ESPECIFICOS

- Analisar e propor uma estrutura e arquitetura dos dados para serem implementados numa aplicação WEB via método dos elementos finitos;

- Estudar a interface gráfica de aplicações desktop e trabalhos disponíveis para implementação desta aplicação WEB;
- Analisar o impacto desta ferramenta na aprendizagem do aluno;

6. REVISÃO DE LITERATURA

Para o desenvolvimento de um sistema web educacional para ensino de métodos dos elementos finitos, faz-se necessário o estudo de alguns conceitos, tais como softwares educacionais, programação orientada a objeto, programação web, javascript, métodos dos elementos finitos e softwares disponíveis.

6.1. SOFTWARES EDUCACIONAIS

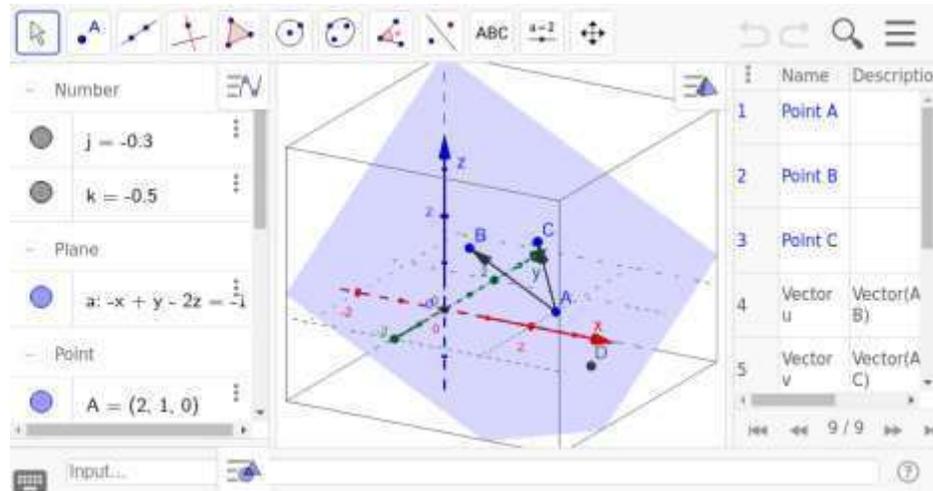
Diante dos grandes avanços tecnológicos, o ensino tem se tornado mais versátil com instrumentos complementares como jogos, softwares e outros métodos dinâmicos. Assim, os professores devem sempre buscar desenvolvimento de uma aula mais motivadora, e que, proporcione uma melhor qualidade na transmissão do conhecimento, para que os alunos compreendam melhor e mais rapidamente um determinado conteúdo.

Portanto, para desenvolvimento lúdico em sala de aula cabe ao professor pesquisar novas ferramentas educacionais de modo em que os alunos possam interagir, realizar simulações e desenvolver protótipos com aplicações educacionais com propósito de atrair a atenção do aluno para o conteúdo ministrado.

Com a tecnologia cada vez mais avançada a informática na educação tende proporcionar aos educadores novas formas de instigar os alunos nas tomadas de decisões, e com autodesenvolvimento pessoal, motivação, foco para aprendizagem com inovação para o ensino por intermédio da tecnologia da informação de acordo com Pacheco (2019).

Entretanto, os recursos tecnológicos não tratam somente do computador ou plataforma de ensino interativa, mais sim de diversas ferramentas para auxiliar o professor na sala de aula tais como, tutoriais, programação, aplicativos, exercícios e práticas, multimídia e Internet, simulação, gamificação e quizz, de acordo com Vieira (2015). Todas as aplicações são utilizadas para que o aluno possa absorver todo conteúdo e gerar interesse de pesquisa, tal como a ferramenta GeoGebra que é um aplicativo de matemática dinâmica que combina conceitos de geometria e álgebra em uma única interface gráfica, conforme a figura 01.

Figura 01: Aplicação da equação vetorial do plano



Fonte: www.geogebra.org

No contexto de softwares para auxílio no ensino e aprendizagem na formação de profissionais, existem diferentes tipos de softwares com objetivo educacional tais como:

- Softwares educativos, visa educar com base em uma reflexão pedagógica.
- Softwares educacionais, tem como finalidade a objetivos específicos seja ele educacional ou uma prática pedagógica.

Desta maneira todo software pode ser considerado educacional, contudo, deve estar inserido no contexto, conforme (LISBOA, 2014 e ROCHA, 2019).

O desenvolvimento de programas e sistemas computacionais voltados à aprendizagem tem sido ainda pouco explorada no ambiente de sala de aula, de acordo com (BERTOLETTI, p. 03, 2003). À vista disto, o uso das tecnologias digitais da informação e comunicação no processo de ensino, possibilita ao aluno o autodesenvolvimento de trabalhos mais completos pedagogicamente.

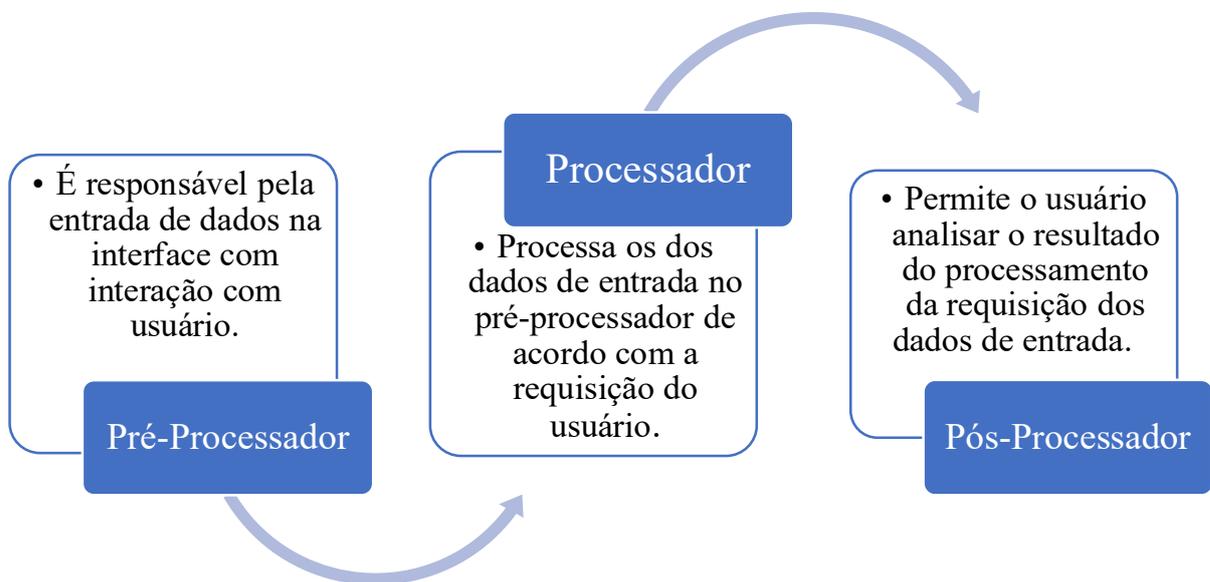
6.2. SOFTWARES NA ENGENHARIA

A engenharia tem evoluindo gradualmente com a tecnologia da informação, com diversas aplicações desde análise estrutural até realidade virtual (RV). Entretanto, os softwares

desenvolvidos para engenharia são pagos com valor de aquisição muito alto, e os gratuitos são apenas para aplicações desktop, além do valor da aquisição ter o custo computacional muito alto para uma análise, e inviabilizando algumas universidades na aquisição para uso educacional, de acordo com Mascarenhas (2020).

O desenvolvimento de programas para análise estrutural se divide em três etapas, sendo pré-processador, processador e pós-processador de acordo com (BOTELHO, 2015; ROCHA, 2018; MOREIRA, 2018) na figura 02 abaixo:

Figura 02: Pré-processador, processador e pós-processador



Fonte: Autor (2022)

As Aplicações podem auxiliar em um estudo prévio das estruturas antes do desenvolvimento de protótipos. Por tanto, esses protótipos virtuais são de extrema importância, possibilitando a simulação de cargas em estruturas física e seus componentes. Vale ressaltar que o protótipo físico é indispensável nas análises, entretanto com uma aplicação há redução de custo de ensaios físicos de corpos de provas e protótipos, de acordo com Costa (2021).

Os softwares utilizados em cálculo de estruturas auxiliam tanto na obtenção de desenvolvimento de análises numéricas como em representações gráficas, sustentando uma compreensão do comportamento da estrutura (BRANCHIER, p.41, 2018). No entanto, os mesmos cálculos feitos de processo manual se tornam inviáveis, já que para resolução de um cálculo demandariam muito tempo.

O profissional tem que acompanhar as inovações tecnológica, visto que o uso de softwares no presente momento se torna indispensável no decorrer do seu dia de trabalho, sendo

um dos requisitos para grandes empresas que o profissional tenha experiência com alguns softwares. Contudo, se faz necessário que a formação profissional nos cursos de graduação e pós-graduação contem não só ensino teórico, e sim pratica em softwares proporcionando experiencia para que possibilite o aumento dos seus resultados de acordo com Nazário (2020).

Levando em consideração que há diversos tipos de aplicações que vem sendo desenvolvida nos últimos anos, com objetivo educacional e gratuito, como são conhecidos *softwares* educacionais, de acordo com Junior (2016), esses softwares auxilia no processo de ensino e aprendizagem, com finalidade que o aluno assimile com mais facilidade o conteúdo das disciplinas mais complexas. Embora, já mencionado aplicações educacionais WEB para método dos elementos finitos com fins educacionais são bastante escassos até a presente dada.

6.3. APLICAÇÕES WEB

Com incessantes avanços na rede mundial de computadores tem viabilizado grandes melhorias na comunicação com aplicações voltada para computação em nuvem, com objetivo de armazenamento e processamento em grandes servidores sem que haja necessidade de um computador com alta capacidade de processamento, de acordo com Soares (2022). Portanto, os servidores tem grande capacidade de processamento por ser computadores mainframe, dessa maneira o com os avanços tecnológico possibilita o trabalho colaborativo de forma remota possibilitando mais acessibilidade ao usuário.

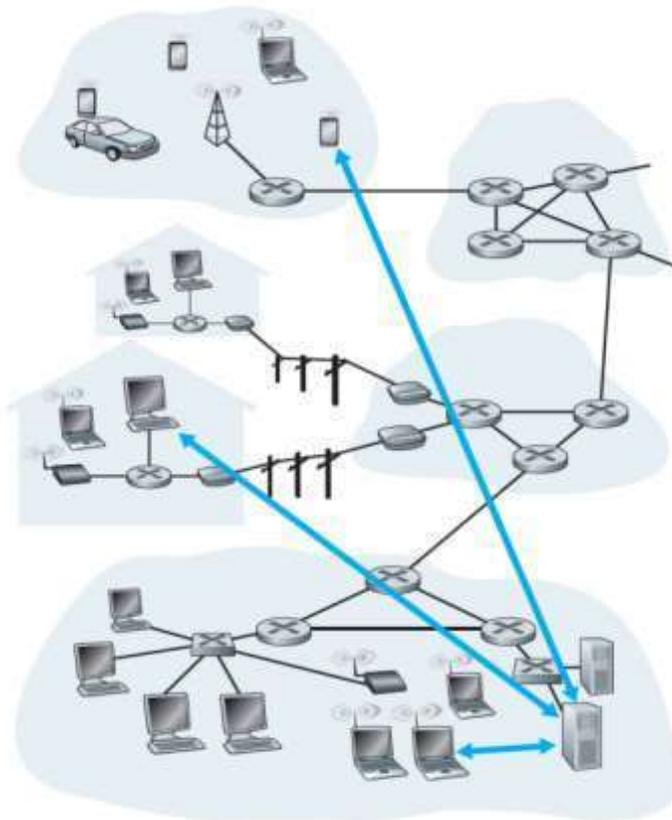
No decorrer da evolução dos softwares e aplicações web é possível identificar o grande avanço por volta do século XX, onde deveria existir uma máquina exclusiva para uma aplicação, chegando a levar dias para realizar um processamento de um cálculo. Contudo, atualmente, com a evolução da computação em nuvem possibilitando a disponibilidade sob demanda de recursos, tais como softwares, armazenamento e processamento de dados, de maneira remota em alta velocidade, as aplicações podem ser usadas de forma remota e com alta eficiência, conforme Camboim (2018). À vista disto, o desenvolvimento de aplicações web tiveram um grande avanço, sendo ela para simulação ou outras opções afins.

Deste modo para o funcionamento dos data center no desenvolvimento de aplicações, se dá em virtude a arquitetura cliente/servidor. Esta arquitetura funciona da seguinte maneira, para o servidor que está hospedada aplicação seja em JavaScript, WebGL ou Bootstrap recebe uma requisição do usuário por meio de URLs e realiza a entrega por meio do protocolo

Hypertext Transfer Protocol (HTTP). Já para o usuário/cliente realizar as requisições ao servidor por meio de URL, conforme a figura 02, de acordo com Oliveira (2017). As mensagens trocadas por esses processos são enviadas passando por camadas, que serão discutidas na próxima seção. Um processo recebe e envia mensagens para a rede através de uma interface chamada de socket, que serve como uma interface de programação entre as camadas da aplicação e de transporte. Assim, as solicitações e as respostas transitam da camada de transporte para a camada de aplicação por meio dos sockets conforme a figura 03. Embora os desenvolvedores tenham limitado acesso à programação da camada de transporte, desfrutam de maior liberdade nas escolhas relacionadas à implementação na camada de aplicação.

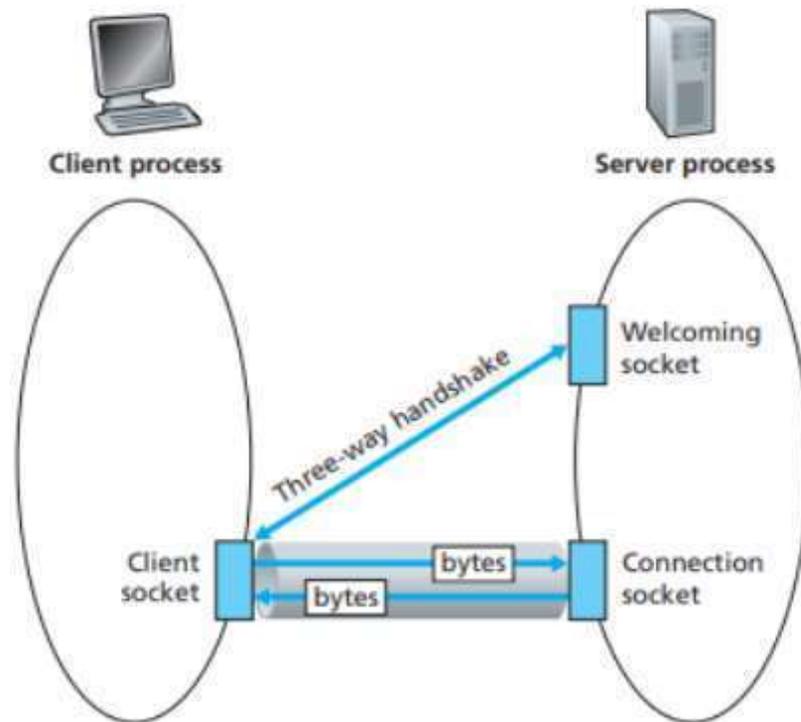
Os Softwares de aplicações web são acessados através de navegadores web, utilizando a infraestrutura da internet para funcionar. Essas aplicações são desenvolvidas com tecnologias web, como HTML, CSS e JavaScript, e podem ser executadas em diferentes dispositivos, como computadores, smartphones e tablets. As aplicações web têm sido amplamente utilizadas em diversos setores devido às suas vantagens e benefícios.

Figura 03: Arquitetura cliente/servidor



Fonte: Soares (2019)

Figura 04: Interface de programação da aplicação socket



Fonte: Kurose & Ross (2013)

Entretanto diversos autores tais como (AZEVEDO, 2018; COSTA, 2018; PACHECO, 2019; BORGES, 2021). Relatam que pouco programadores sabem da vulnerabilidade das aplicações web, apesar disso os mesmos autores frisam que as aplicações web mais vulneráveis são aquelas que tem banco de dados integrado. Contudo, as aplicações web no seu desenvolvimento tem todos os critérios de segurança possibilitando ao cliente um acesso seguro com alta capacidade de processamento de maneira remota.

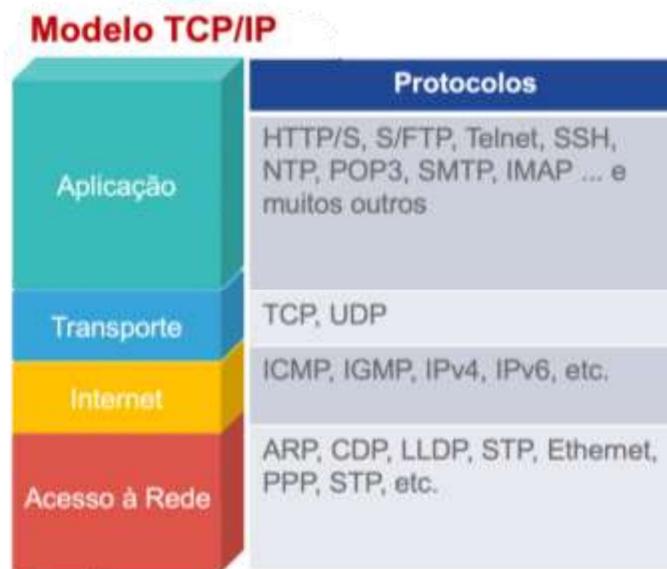
6.3.1. ARQUITETURA DO TCP/IP

Os modelos de rede são estruturados em arquiteturas de camadas, sendo que cada camada possui protocolos dedicados que operam sobre ela. Essas camadas são dispostas de maneira hierárquica, formando uma pilha na qual cada camada superior se beneficia dos serviços oferecidos pela camada imediatamente inferior, de acordo com De Souza (2023).

Igualmente é chamado de pilha de protocolos, o modelo TCP/IP apresenta uma subdivisão em quatro camadas distintas: aplicação, transporte, internet e enlace. Os protocolos variam nas camadas de aplicação. A camada de aplicação, sendo a camada mais próxima do

usuário, é considerada de mais alto nível. Em contraste, a camada de enlace é considerada de mais baixo nível, pois está mais distante do usuário, de acordo com Jabs (2015), conforme a figura 04 abaixo. Vale salientar que os protocolos e camadas citados aqui neste trabalho são as básicas que precisa para o funcionamento da aplicação e serão todas detalhadas nos próximos parágrafos.

Figura 05: Aplicação TCP/IP e seus protocolos



Fonte: JABS (2015)

6.3.2. APLICAÇÃO

A camada aplicação possui diversos protocolos como tais como HTTP/S, SSH, NTP, POP3, SMTP, IMAP e TELNET, Esses protocolos da camada de aplicação desempenham funções específicas para facilitar diferentes tipos de comunicação entre os usuários e os serviços da rede. O NTP sincroniza relógios entre sistemas, enquanto o POP3, SMTP e IMAP são protocolos de e-mail com diferentes abordagens para acesso e gerenciamento de mensagens. O TELNET, por sua vez, oferece uma maneira de acessar remotamente servidores e sistemas. Cada um desses protocolos contribui para a diversidade e eficiência das interações na camada de aplicação.

De acordo com Barros (2022), O protocolo HTTP estabelece a metodologia pela qual as aplicações clientes formulam e enviam solicitações aos servidores, além de especificar a maneira como os servidores respondem, transmitindo informações de volta para as aplicações clientes. Esse processo é essencial para a comunicação eficiente e a transferência de dados na World Wide Web.

O HTTP opera sobre o TCP como seu protocolo de transporte. Primeiramente, o cliente estabelece uma conexão com o servidor, realizando um processo conhecido como "handshake", o qual é essencial para a comunicação subsequente. Após essa etapa, o cliente pode efetuar suas requisições, enquanto o servidor responde a essas solicitações. O HTTP adota métodos padrões para as requisições, incluindo GET, POST, HEAD, PUT e DELETE, oferecendo uma gama de opções para interações específicas entre clientes e servidores. Essa abordagem padronizada contribui para a consistência e eficiência na comunicação web, segundo Ramos Junior (2022).

O método GET é utilizado para solicitar dados de um recurso específico no servidor. É amplamente utilizado para recuperar informações, e a solicitação não deve ter efeitos colaterais no servidor. O método POST é utilizado para enviar dados para serem processados para um recurso específico no servidor. É comumente usado para enviar informações de formulários e pode ter efeitos colaterais no servidor, como a atualização de um banco de dados. O método HEAD é semelhante ao GET, mas é usado para obter apenas os cabeçalhos da resposta, sem o corpo da mensagem. É útil para obter informações sobre o recurso sem transferir todo o conteúdo, economizando largura de banda. O método PUT é utilizado para enviar dados para serem armazenados em um recurso específico no servidor. Ele substitui completamente o conteúdo do recurso existente ou cria um novo recurso se não existir. O método DELETE é utilizado para solicitar a remoção de um recurso específico no servidor. Ele solicita que o servidor remova o recurso identificado pela URI fornecida.

6.3.3. TRANSPORTE

O transporte sendo uma das camadas que possui dois protocolos sendo eles UDP e TCP a mesma desempenha um papel crucial na comunicação ponto a ponto entre sistemas finais. Nessa camada, dois dos protocolos mais proeminentes são o TCP (*Transmission Control Protocol*) e o UDP (*User Datagram Protocol*). Para Hecht (2021) o TCP é reconhecido por proporcionar uma comunicação confiável, estabelecendo uma conexão robusta que garante a entrega precisa e sequencial dos dados, sendo especialmente útil em situações em que a integridade e a ordem dos dados são críticas.

Em contrapartida, o UDP destaca-se pela sua orientação a serviços mais ágeis, sendo amplamente empregado em aplicações que demandam transmissão de dados em tempo real, como em sistemas de streaming, onde a velocidade de transmissão é prioritária sobre a garantia absoluta de entrega de todos os pacotes. Essa distinção entre TCP e UDP oferece flexibilidade

na escolha do protocolo, permitindo adaptar a abordagem de comunicação de acordo com as necessidades específicas de cada aplicação. É importante ressaltar que, embora o UDP ofereça benefícios em termos de velocidade, ele não garante a entrega ou a ordem dos pacotes, sendo necessário que as aplicações que o utilizam lidem com eventuais perdas ou desordens, se necessário.

6.3.4. INTERNET

Camada internet ela é responsável pelos seguintes protocolos tais como ICMP, IGMP, IPv4 e IPv6 entre outros. Portanto a mesma tem uma papel fundamental na interconexão de diferentes redes, facilitando o roteamento e a transmissão eficiente de dados em escala global. O protocolo central nesta camada é o IP (Internet Protocol), que desempenha um papel crucial ao endereçar e rotear pacotes de dados através da vasta infraestrutura da Internet. Essa camada é vital para garantir que os dados se movam de forma eficaz de uma origem para um destino, independentemente das complexidades da rede interconectada.

Podemos citar os exemplos mais proeminentes do protocolo IP incluem o IPv4 (Internet Protocol version 4) e o IPv6 (Internet Protocol version 6). O IPv4, por ser a versão mais amplamente utilizada, utiliza endereços de 32 bits, o que limita o número total de endereços disponíveis. Devido ao crescimento exponencial da Internet e à necessidade de mais endereços únicos, o IPv6 foi desenvolvido. O IPv6 utiliza endereços de 128 bits, proporcionando um número significativamente maior de endereços únicos e, assim, atendendo às crescentes demandas de endereçamento na era digital.

Esses protocolos IP desempenham um papel vital no funcionamento cotidiano da Internet, permitindo a comunicação global, o acesso a recursos distribuídos e a troca de informações entre sistemas interconectados. O roteamento eficiente de pacotes através da camada de internet é essencial para a conectividade e o funcionamento harmonioso da rede mundial.

6.3.5. ACESSO À REDE

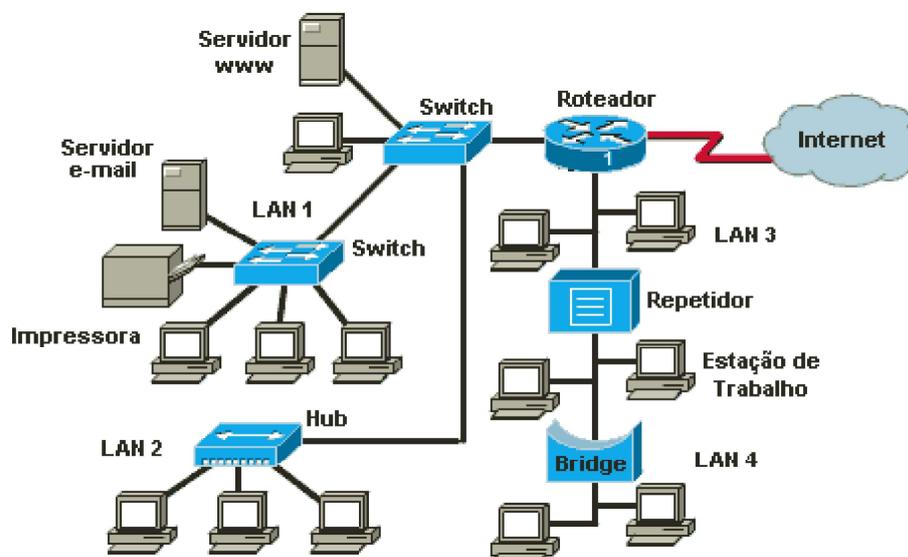
A camada de enlace de dados assume um papel crucial ao garantir a transmissão confiável de dados por meio de diversos meios físicos, como cabos, fibra óptica e tecnologias sem fio. Esta camada, essencial para o funcionamento eficiente das redes, é subdividida em

duas subcamadas distintas: a LLC (Logical Link Control), encarregada do controle lógico do enlace, e a MAC (Media Access Control), responsável por regular o acesso ao meio físico.

A subcamada LLC desempenha funções vitais na transmissão de dados, incluindo o controle de fluxo, a detecção e correção de erros, e a retransmissão de pacotes para assegurar a integridade da comunicação segundo Souza (2019). Enquanto isso, a subcamada MAC gerencia a comunicação no nível físico, supervisionando o acesso ao meio de transmissão compartilhado, garantindo que dispositivos conectados possam transmitir e receber dados de maneira ordenada e eficiente.

Um exemplo visual dessa interação pode ser observado em uma rede local que utiliza Ethernet com fio. Considere uma imagem de uma rede com vários computadores interconectados por meio de cabos Ethernet conforme a figura 05 abaixo. A camada de enlace de dados, representada pelos protocolos Ethernet, é a responsável por garantir que os dados sejam encapsulados adequadamente, transmitidos sem colisões e recebidos corretamente pelos dispositivos conectados.

Figura 06: Exemplo de Rede Ethernet



Fonte: DOS SANTOS PINHEIROS (2004)

Além disso, em ambientes onde a comunicação é realizada sem fios, como redes Wi-Fi, a camada de enlace de dados também desempenha um papel crítico ao coordenar o acesso ao meio físico, possibilitando uma transmissão eficiente de dados entre dispositivos conforme Ilário (2020). Dessa maneira, a camada de enlace de dados é um componente vital para a transmissão confiável de dados, independente do meio físico utilizado, contribuindo para a eficácia e integridade das comunicações em redes.

6.4. JAVASCRIPT

JavaScript uma das linguagens de programação mais utilizada desde a data da sua criação na década de 90, com cerca de 16,4 milhões de desenvolvedores conforme ZDNET (2021) na figura 03. A mesma possibilita desenvolver páginas web complexa, como gráfico e mapas 2D/3D com animações interativas ou softwares com interação com banco de dados. Entretanto a mesma é utilizada com outras duas linguagens iniciais, sendo HTML e CSS.

Figura 07: Linguagens de programação mais utilizadas em 2021



Fonte: www.zdnet.com

A linguagem de programação Java embora tenha o nome parecido com JavaScript e seja registrado pela mesma empresa Oracle, ambas têm sintaxes diferentes onde JavaScript o foco principal é em desenvolvimento de páginas web, jogos, sistemas de simulação gráfica e com interação pelo navegador em qualquer dispositivo seja ele celular, desktops ou tablets, conforme MDN web docs (2022).

Com diferentes níveis em linguagem de programação JavaScript, é uma linguagem de alto nível com muitos comandos complexo e com uma diversificação de termos, a mesma é baseada na arquitetura cliente/servidor com a flexibilidade de acesso em navegadores web, permitindo que altere o HTML com CSS pelas *Application Programming Interfaces* APIs. Além disto, é hoje a linguagem documentada e utilizada por desenvolvedores sendo assim a

mais versátil. Com ela é possível utilizar os três principais paradigmas de programação: procedural, orientado a objetos e funcional de acordo com Teixeira (2018).

O paradigma programação orientada a objetos POO tem como objetivo a reutilização do código fonte replicando as classes e funções que pode ser utilizado em diferentes momentos da aplicação, conseqüentemente reduzindo o número de erros e facilitando a sua manutenção. O mesmo é fundamentado em quatro pilares para o seu funcionamento sendo eles Abstração, encapsulamento, herança e polimorfismo, segundo Pinheiro (2019). Embora, ao usar a programação não exista ainda uma forma formal de se utilizar abstração, assim sendo para usar a abstração tem sido feito arranjos simplificados.

7. METODOLOGIA

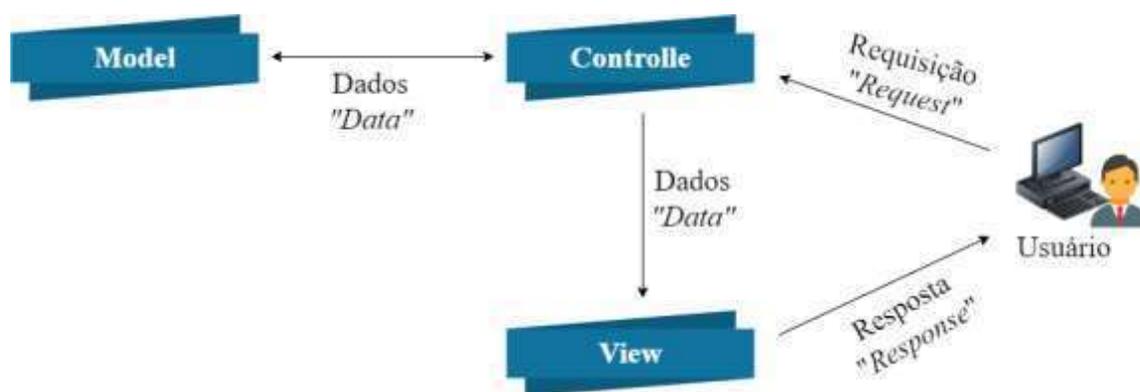
Para realização desta pesquisa foi realizado uma revisão sistemática de literatura para entender a necessidade de um software para ensino de métodos elementos finitos conforme o capítulo 01 desta pesquisa, A revisão da literatura permitiu explorar as aplicações web já existentes no contexto educacional, analisando suas vantagens e desafios. Estudos relevantes foram revisados para embasar a decisão de desenvolver um software web educacional, demonstrando a eficácia desse tipo de abordagem no ensino e aprendizagem.

Nesta pesquisa foi desenvolvida uma prova de conceito de um aplicativo web para análise de pórticos planos. Para isso, foi escolhido o ambiente de desenvolvimento integrado (IDE), o Visual Studio Code (VSCode), na versão 1.68. Este ambiente, é um software gratuito desenvolvido pela *Microsoft* para os sistemas operacionais Linux, Windows e macOS, tendo como foco o desenvolvimento web com *javascript*, *typescript* e *Node.js*. Entretanto, a mesma tem suporte para desenvolvimento em outras linguagens de programação, além disso, é uma IDE totalmente customizável possibilitando alteração do ambiente de desenvolvimento.

A linguagem de programação que está sendo utilizada é *JavaScript*, por ser uma linguagem para desenvolvimento de aplicações de alto nível em web, baseada na arquitetura cliente/servidor. Para a padronização do desenvolvimento da aplicação está sendo utilizado o *Model-View-Controle* (MVC) conforme a figura 05, que visa na organização do código-fonte possibilitando que outro programador possa entender todo código. O padrão MVC possui sua divisão em camadas sendo elas:

- Model: A responsabilidade do model é representar a regra de negócio. Também é responsável pelo acesso e manipulação dos dados na sua aplicação.
- View: É responsável pela interface que será apresentada, mostrando as informações do model para o usuário.
- Controlle: Fica encarregado por ligar o model e a view, fazendo com que os models possam ser repassados para as views e vice-versa.

Figura 08: Padrão MVC

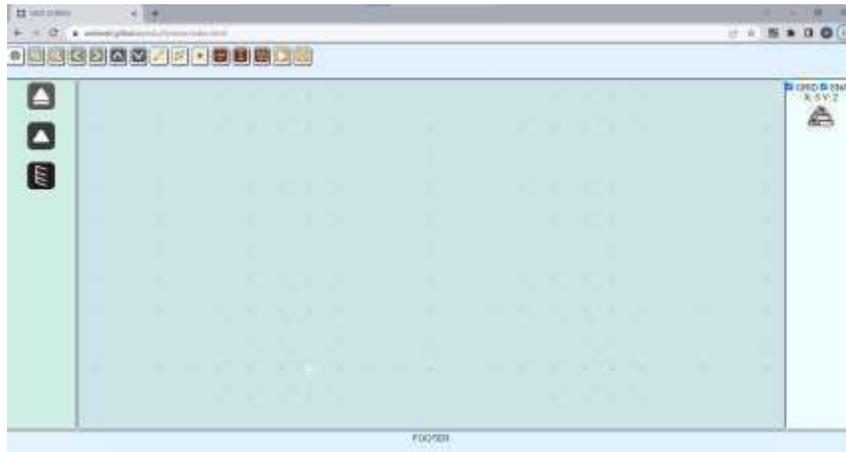


Fonte: Autor (2022)

Um esquema de layout da aplicação pode ser visto conforme a figura 06. Entretanto, para o desenvolvimento destas funcionalidades teve que ser criado classes, atributos e métodos. A Aplicação se divide nos seguintes arquivos *JavaScript* linguagem de alto nível, HTML linguagem de marcação e CSS que é o estilo da aplicação.

Os arquivos JavaScript são compostos pelo padrão *Model-View-Controller*, funções para renderização, criação da câmera e funções de todos os botões, funções do teclado. O HTML dispõe somente de um arquivo chamado de *index*, neste arquivo é realizado todas as comunicações com os outros arquivos como JS, P5js e CSS, no mesmo é adicionado o nome da aplicação, linguagem entre outras funcionalidades. Já o CSS é o arquivo responsável pelas cores, estilo de letras, essa formatação pode ser aplicada diretamente nas tags ou ficar contido dentro das tags `<style>`.

Figura 09: Layout da aplicação



Fonte: Autor (2022)

Para o uso da ferramenta para análise de elementos p3rticos planos com fins educacional, ser3a necess3rio que o usu3rio tenha acesso a um dispositivo com internet, seja ele celular, tablete, computador ou notebook/netebook. O mesmo basta acessar um navegador e digitar o endereço web <https://webmef.github.io/edu/> logo ap3s o usu3rio ter3 acesso a p3gina de informaço3es sobre os autores, vale salientar que todas as p3ginas tem o um bot3o que redirecionar o mesmo para pagina de an3lise de elementos p3rticos planos, para desenvolvimento dos elementos tem informaço3o dispon3vel no capitulo dois e na p3gina do *github* no link a seguir <https://github.com/webmef/edu> com c3digo fonte para contribuiço3o na mesma ferramenta.

Entretanto, ainda falta implementaço3o das seguintes funço3es tais como, implementaço3o de elementos 3D, criaço3o de cadastro de usu3rios com intuito que cada usu3rio possa ter seu ambiente de desenvolvimento pr3prio, atualizaço3o para implementaço3o de recursos gr3ficos, criaço3o da funço3o da apagador, criar um tipo de arquivo que seja poss3vel o usu3rio realizar upload e ediço3o do mesmo, possibilitar mais opço3es de download, geraço3o de grid com analise especifica de MEF e desenvolvimento de um execut3vel.

REFER3NCIAS BIBLIOGR3FICAS

AZEVEDO, Daniel Galv3o de. **Um estudo sobre ferramentas de busca de vulnerabilidades**

em aplicações web. 2018. Trabalho de Conclusão de Curso. Universidade Federal do Rio Grande do Norte.

BARROS, Abimael Silas. Análise de desempenho dos protocolos coap, mqtt-sn e http em redes IoT. 2022.

BERTOLETTI, Ana Carolina et al. Educar pela Pesquisa—uma abordagem para o desenvolvimento e utilização de Softwares Educacionais. **RENOTE**, v. 1, n. 2, 2003.

BONIATI, Bruno Batista; SILVA, Teresinha Letícia. Fundamentos de Desenvolvimento Web. **Frederico Westphalen, RS. Acesso em**, v. 1, 2013.

BORGES, Vitor OCN; UCHÔA, Joaquim Q. FuzzingTool: Ferramenta para Testes de Intrusão em Aplicações Web. In: **Anais do XXI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais**. SBC, 2021. p. 391-396.

BOTELHO, Guilherme Garcia; DA FONSECA, F. T.; PITANGUEIRA, R. L. S. Sistema computacional orientado a objetos para análises acopladas termo-estruturais pelo método dos elementos finitos. In: **Proceedings of the XXXVI Ibero-Latin American Congress on Computational Methods in Engineering-CILAMCE**. 2015.

BRANCHIER, Henrique Scalcon. **Contribuições dos softwares na aprendizagem de análise e cálculo de elementos estruturais**. 2018. Trabalho de Conclusão de Curso.

CAMBOIM, Kádna; ALENCAR, Fernanda MR. Requisitos não Funcionais e Sustentabilidade para Computação em Nuvem: uma Revisão Sistemática da Literatura. **WER**, 2018.

COSTA, Leonardo Ribeiro. Protótipos estruturais com uso de impressora 3D: proposta de intervenção pedagógica com metodologias ativas para desenvolvimento de novas habilidades profissionais. 2021.

COSTA, Pablo V. et al. Nível de conhecimento de desenvolvedores sobre segurança em aplicações web: Pesquisa e análise. In: **Anais da V Escola Regional de Sistemas de Informação do Rio de Janeiro**. SBC, 2018. p. 92-99.

DA SILVA NAZÁRIO, Lanna Celly; BENTO, Fernanda Beatriz da Silva Bento. A importância do uso de softwares de gerenciamento de projetos na formação do aluno de engenharia civil. **Brazilian Journal of Development**, v. 6, n. 8, p. 54668-54677, 2020.

DE OLIVEIRA, Sérgio. **Internet das coisas com ESP8266, Arduino e Raspberry PI**. Novatec Editora, 2017.

DE SOUZA, Joao Pedro Bernardo; PANTOJA, Carlos Eduardo; LAZARIN, Nilson Mori. Comunicação entre SMA Embarcados: Uma Arquitetura Baseada em Protocolos da Camada de Aplicação. 2023.

DOS SANTOS PINHEIRO, José Maurício et al. Ameaças e ataques aos sistemas de informação: Prevenir e antecipar. **Cadernos UniFOA**, v. 3, n. 5, p. 11-21, 2007.

ILÁRIO, Lucas Rodrigues. Contribuições para integração da indústria 4.0 nas redes elétricas inteligentes. 2020.

Instituto Federal Goiano. **Apoio e Modelos**. Página inicial. Disponível em: <<https://ifgoiano.edu.br/home/index.php/ultimas-noticias-morrinhos/2-uncategorized/7603-apoio-emodelos.html>>. Acesso em: 18 de agosto de 2021.

JABS, Darney. Placa microcontrolada para controle supervisorio com interface web. 2015. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

JÚNIOR, Ozonias Brito; AGUIAR, Yuska; TAVARES, Tatiana. Abordagens para Avaliação de Softwares Educativos e sua Coerência com os Modelos de Qualidade de Software. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. 2016. p. 270.

LISBOA, Luís Eduardo; NUNES, Jorge Adriano Carneiro. SOFTWARES EDUCATIVOS. **Maiêutica-Ensino de Física e Matemática**, v. 2, n. 1, 2014.

MASCARENHAS, Fernando Júnior Resende; CHRISTOFORO, André Luis; CARVALHO, Roberto Chust. A utilização de um software livre para determinação dos esforços internos de cargas móveis em vigas de pontes de concreto. **HOLOS**, v. 3, p. 1-18, 2020.

MDN. **O que é JavaScript? - Aprendendo desenvolvimento web** | MDN. Disponível em: <https://developer.mozilla.org/ptBR/docs/Learn/JavaScript/First_steps/What_is_JavaScript>. Acesso em: 1 jul. 2022.

MOREIRA, Marco Aurélio de Oliveira et al. **Obtenção do desenho geométrico de objetos a partir de imagens para utilização no método de elementos finitos**. 2020

MOREIRA, Renata Nicoliello; PITANGUEIRA, Roque Luiz. Sistema gráfico interativo para ensino de análise estrutural através do método dos elementos finitos. **Revista de Ensino de Engenharia**, v. 37, n. 1, 2018.

NETO, Alvaro M. Delgado; GUIMARÃES, Denise C.; GOMES, Gilberto. BEMLAB2D: GUI modeling, visualization and analysis with boundary element—an application in elastostatic problems.

PACHECO, Erica Farias. Utilizando o software geogebra no ensino da matemática: uma ferramenta para construção de gráficos de parábolas e elipses no 3 ano do ensino médio. **Debates em Educação**, v. 11, n. 24, p. 197-211, 2019.

PACHECO, Felipe Guimarães. **Análise das técnicas de segurança do framework laravel contra ataques as aplicações web**. 2019. Trabalho de Conclusão de Curso. Brasil.

PINHEIRO, Rogélio Carpes et al. Aplicativo de desdobramento das funções da qualidade (QFD) utilizando conceitos de programação orientada a objetos. **Blucher Design Proceedings. Blucher**, 2019.

RAMOS JÚNIOR, Levi da Silva. RequestBERT-BiLSTM: detecção de ataques em requisições HTTP sem log parser. 2022. Dissertação de Mestrado. Universidade Federal de Pernambuco.

ROCHA, Pedro Luiz Ribeiro. Desenvolvimento de um pré-processador para análise de estabilidade de perfis estruturais de paredes finas. 2018.

ROCHA, Paul Symon Ribeiro; RAMOS, Carlos Vieira; BRASIL, Tainara Antunes. **A utilização de softwares no ensino de matemática para ensino fundamental e médio**. In: Anais do IV Congresso sobre Tecnologias na Educação. SBC, 2019. p. 40-49.

SOARES, Rodrigo Lucas. **Desenvolvimento de uma aplicação web para modelagem colaborativa**. 2022. Dissertação de mestrado. PUC-Rio.

SOUZA, Fernando Henrique Nascimento de. Rede sensível ao tempo: um estudo do mapeamento sistemático. 2019.

TEIXEIRA, Fábio; SILVA, Régio. Jogo Digital de Memória para Crianças com Baixa Visão utilizando Programação Orientada a Objetos. 2018.

VIEIRA, Fábila Magali Santos. Classificação de softwares educacionais. **Mídias na Educação**, v. 12, n. 1, p. 1-3, 2015.

**CAPÍTULO 1 – ARTIGO 1
(FOLHA DE ROSTO)**

Título do Artigo: Uma prova de conceito de um aplicativo web para análise de pórticos planos: Uma revisão sistemática

Qualis: A4

ISSN: 2316-9451

Área: Ciência da computação

Nome da revista: Journal of Interdisciplinary Studies on Science and Informatics

Uma prova de conceito de um aplicativo web para análise de pórticos planos: Uma revisão sistemática

A proof of concept of a web application for analyzing flat frames: A systematic review

RESUMO

O método dos elementos finitos é muito utilizado na engenharia civil, principalmente no estudo estrutural. Porém, não é um assunto simples e rápido de ser ensinado, e requer bom domínio dos alunos na visualização das geometrias e modelos. Existem vários *softwares* comerciais e alguns estudantis. A maioria dos *softwares* para refinamento de malhas, são para aplicações para desktops e tem seu custo por licença bastante alto chegando até a 2000USD, dificultando pequenas instituições de ensino realizar simulações. Neste sentido, a interface gráfica do software é de extrema importância para melhor ensinar este método. Então, este estudo visa realizar uma análise bibliográfica sobre quais, o estado da arte de aplicações gráficas para método dos elementos finitos, sejam via web ou via desktop. Para o desenvolvimento desta revisão foi utilizado o software StArt, desenvolvida pelo Laboratório de Pesquisa em Engenharia de Software (LaPES) da Universidade Federal de São Carlos (UFSCar), possuindo três etapas sendo elas o protocolo que é o planejamento do estudo, execução que é responsável pela identificação, seleção e extração dos estudos, e última etapa sumarização que é composto pelos gráficos e resultado da revisão. Os resultados mostram que há uma quantidade diversificada de *softwares* web e desktops para auxílio do ensino em diversas áreas. Entretanto para o ensino de métodos de elementos finitos com engenharias de estruturas em web que seja gratuito e acessível não foi encontrado, os estudos encontrados de *softwares* para MEF são aplicação desktops e sem atualizações, desta maneira ocasiona uma carência dessa aplicação.

Palavras chaves: sistema educacional; MEF; educação baseada na web; aprendizagem;

ABSTRACT

The finite element method is widely used in civil engineering, especially in structural studies. However, it is not a simple and quick subject to be taught, and requires a good command of the students in the visualization of geometries and models. There are several commercial and some student software programs. Most of the mesh refinement software is for desktop applications and its cost per license is very high, reaching up to 2000USD, making it difficult for small educational institutions to perform simulations. In this sense, the graphical interface of the software is of extreme importance to better teach this method. Therefore, this study aims to perform a literature review on what is the state of the art of graphical applications for the finite element method, either via web or desktop. For the development of this review it was used the software StArt, developed by the Software Engineering Research Laboratory (LaPES) of the Federal University of São Carlos (UFSCar), having three stages being them the protocol that is the study planning, execution that is responsible for the identification, selection and extraction of the studies, and last stage summarization that is composed by the graphics and the review result. The results show that there is a diverse amount of web and desktop software to aid teaching in several areas. However, for the teaching of finite element methods with structural engineering on the web that is free and accessible was not found, the studies found of FEM software are desktop applications and without updates, thus causing a lack of this application.

Keywords: educational system; FEM; web-based education; Learning;

1 INTRODUÇÃO

As Tecnologias Digitais da Informação e Comunicação (TDIC's), tem provocado um grande impacto direto na sociedade no modo de comunicar chegando estimular o desenvolvimento de até novos hábitos. As tecnologias da informação atualmente estão presente em diversas áreas desde construção civil, saúde, indústria e educação.

Para o ensino, as tecnologias têm grande importância desde as séries iniciais até pós-graduação, pelo motivo de ter diversos conteúdos de fácil acesso desde vídeo, livros principalmente pelo fácil acesso por meio de computadores, celulares e tablets de qualquer localidade que tenha acesso a internet, possibilitando a simulação de exercícios, Da Silva Neto (2013).

O professor é responsável pelo acompanhamento e planejamento de atividades, com o uso de recursos tecnológicos dentro e fora da sala de aula, que os alunos não seja apenas mero receptor e sim que tenha interação direta com conteúdo (MARCELINO, 2017; CARDOSO, 2018). Apesar de que há diversos autores relatando a importância das tecnologias ainda há professores que não trabalha com a mesma por falta de formação continuada.

O uso das novas técnicas de ensino através de recursos tecnológico como ferramentas educacionais, surge como um auxílio para o processo de ensino em todas as áreas na educação (OLIVEIRA, 2018; COSTA, 2019). Com todo esse método não irá facilitar a profissão do docente e sim, contribuir para melhorar sua capacidade de repassar o conteúdo de forma que todos possam compreender (OLIVEIRA, 2018; DE OLIVEIRA TESSARO, 2019; PASTORIO, 2021).

Entretanto com diversos estudos comprovando que as tecnologias têm grande importância para o ensino, vem sendo desenvolvidos diversos *softwares*, Mendes (2018), tal como, desenvolveu um programa de elementos finitos em ambiente aberto, com intuito que surjam diversas contribuições para aplicação e atualizações futuras. Santos (2018), aponta que a abordagens distintas para educação em computação gráfica não convencionais de alguma forma tentam auxiliam nos desafios dentro da sala de aula. Slobodyanyuk (2018), destaca a importância dos sistemas informáticos no processo de formação e estudos de gráficos na engenharia. Song (2017), apontou em sua pesquisa por meio de questionário a expectativa e necessidade de *softwares* de simulação baseado em métodos dos elementos finitos (MEF). Mcdowell (2016), destacou o uso de software educacional online no ensino e de farmacêuticos na distribuição e registro de medicamentos chegando a ganhar dois prêmios na *Reimagine Education* é uma conferência e competição global, aberta a inovadores educacionais de todo o mundo.

Desta maneira, os estudos relacionados aos programas de computador e (MEF), surgiram na década de (1945) por Richard Courant ampliando o trabalho do Walter Ritz que utilizou (MEF) para determinar uma solução aproximada de problemas em mecânica dos sólidos. No entanto os *softwares* de (MEF) só ganhou popularidade em 1960 onde ele pode ser utilizado em diversas aplicações tais como medicina, odontologia, engenharia civil, mecânica e aeronáutica. Colyer (1985), menciona que para o uso de um software de geração de malhas em MEF para ser utilizado era necessário somente uma máquina exclusiva para aquele sistema. No entanto, com a evolução das tecnologias um desktop pode ser utilizado para diversos serviços simultâneo.

Diversos autores realizaram estudos sobre o uso de (MEF) com computação gráfica para refinamento de geometrias complexa em infinitas geometrias menores que podem ser reconhecidas facilmente (COLYER, 1985; FONSECA, 2006; ASTEJO'N, 2010; RUBIO, 2012; MOREIRA, 2018; KANNAN, 2020; RIGGIO, 2021). Embora que haja diversos *softwares* para refinamento de (MEF) boa parte deles se faz necessário a instalação em desktops ou notebooks tendo necessidades de espaço no disco rígido e licença que podem chegar até 2000USD.

Contudo os *softwares* web, não necessita de processamento e espaço em disco rígido no computador do usuário devido que todo o processamento de informação é em um servidor remoto. Para esse proposito o usuário não vai precisar de um *hardware* ou *software* específico para utilizar o sistema web, sendo preciso somente de acesso à internet. Desta maneira o estudante não irá precisar ter que ir ao laboratório da instituição de ensino para realizar uma simulação ou resolução de exercícios.

O presente estudo teve como objetivo a análise do uso da computação gráfica para plataforma web aplicada a engenharia de estruturas. Com foco nas buscas por trabalhos relacionado a *softwares* web para geração de malhas e *softwares* educacional do período de 1980 até 2022 por meio da revisão sistemática de literatura.

1.2 MÉTODO

Nesta seção é abordado os procedimentos metodológicos utilizados na revisão sistemática, com intuito de alcançar o objetivo da pesquisa, que é a análise do uso da computação gráfica para plataforma web aplicada a engenharia de estruturas. E pergunta de pesquisa, se seria possível implementar métodos de computação gráfica aplicada aos métodos de elementos finitos para plataforma web aplicada a engenharia de estruturas?

À priori para o desenvolvimento desta revisão sistemática da literatura (RSL), foi utilizado o *software State of the Art through Systematic Review (StArt)*, com a versão 3.0.3 BETA, que inclui as seguintes atualizações tais como frequência de palavras-chave/nuvem de palavras, estratégia SCAS para seleção inicial semiautomática, retrocesso em bola de neve, colaboração e comunidade StArt Online. Esta versão tem alguns bugs corrigidos (como o "save and next" e os colaborativos) e melhorias, como apoio no tratamento de dados com finalidade de evidenciar, estudos relacionados ao uso da computação gráfica para plataforma web aplicada a engenharia de estruturas.

A ferramenta StArt ela dispõe das seguintes etapas para o desenvolvimento da revisão sistemática, sendo planejamento, execução e sumarização. No planejamento é uma das etapas mais importantes onde é elaborado um protocolo que contém o objetivo de pesquisa, critérios de seleção, palavras chaves, pergunta de pesquisa e bases de dados. Desta maneira em uma revisão sistemática, na qual tem o protocolo planejado antecipadamente na busca de minimizar os erros na condução e investigação do estado da arte do objetivo da pesquisa Dermeval (2019).

O protocolo para esta pesquisa foi dividido em três etapas: fase de planejamento, fase de execução, e, por fim fase de extração, como segue:

- Fase de planejamento: Onde são colocadas as palavras-chaves e encontrados os artigos que tem potencial de serem usados nesta revisão.
- Fase de execução: Nesta etapa são excluídos os artigos:
 1. Duplicados em várias bases;
 2. Se após a leitura dos resumos e abstracts o artigo não possui correlação com o tema;
 3. Se após a leitura do artigo completo não possui correlação com o tema;
- Fase de extração: os artigos podem ser excluídos por: falta de abstract, artigo não completo, e-book e estudos que não correlacionava diretamente com a linha de pesquisa.

Nesta primeira etapa do desenvolvimento deste estudo, foi montado um protocolo minuciosamente com objetivo, pergunta de pesquisa, palavras chaves, critérios de seleção de trabalhos, idiomas, métodos de pesquisa, critérios de qualidades, formulários de extração, bases de dados e critérios de aceitação e exclusão.

Desta maneira foi cadastrado oito bases de buscas no protocolo, sendo elas ACM (ACM Digital Library), Springer (Springer Nature Group), IEEE (IEEE Xplore Digital Library), Cornell (Cornell University Library), Scielo (Scientific Electronic Library Online), Microsoft

Academic (Microsoft Academic Search), Periódicos CAPES (Portal de Periódicos da CAPES), Scopus (Scopus Preview). As bases de dados são diversificadas contendo um maior número de trabalhos relacionado ao objetivo de pesquisa.

Após a seleção das bases de dados, foi desenvolvido *strings* de busca com palavras chaves para cada base de dados conforme a tabela 01, de acordo com objetivo de pesquisa sendo que estudos que não correlacionam com ele mesmo não era selecionado. Entretanto, vale ressaltar que nas bases de dados IEEE, Scielo e Cornell teve *strings* com termos inferior devido que com a quantidade de palavras não encontrava trabalhos relacionado ao tema de pesquisa.

Tabela 01: *Strings* e bases de dados utilizado no estudo

Bases de dados	String de busca
ACM	“computer graphics” AND “structure” AND “engineering” AND “FEM” OR “finite element method” OR “frame” AND “pré-processador” AND “Delaunay triangulation” AND “mesh” OR “grid” AND “Generation” AND “canvas” AND “mouse event” NOT “time” NOT “fluid” NOT “Multiscale” NOT “multiscale” NOT “thermal” NOT “dynamic” AND “educational” OR “education” AND “framework” AND “graphic program” AND “civil engineer”
IEEE	“computer graphics” AND “structure” AND “engineering” AND “FEM” OR “finite element method” OR “frame” AND “Delaunay triangulation” AND “mesh” OR “grid” AND “Generation” AND “canvas” AND “mouse event” NOT “time” NOT “fluid” NOT “Multiscale” NOT “multiscale” NOT “thermal” NOT “dynamic” AND “educacional OR education” AND “framework” AND “object orientation”
Springer	“computer graphics” AND “structure” AND “engineering” AND “FEM” OR “finite element method” OR “frame” AND “Delaunay triangulation” AND “mesh” OR “grid” AND “Generation” AND “canvas” AND “mouse event” NOT “time” NOT “fluid” NOT “Multiscale” NOT “multiscale” NOT “thermal” NOT “dynamic” AND “educational” OR “education” AND “framework”

Cornell University	<i>“computer graphics” AND “structure” AND “engineering” AND “FEM” NOT “fluid” NOT “Multiscale” AND “Delaunay triangulation” AND “mesh” OR “grid” AND “Generation” AND “canvas”</i>
Scielo	<i>“computer graphics” AND “structure” AND “engineering”</i>
Microsoft Academic	<i>“computer graphics” AND “structure” AND “engineering” AND “FEM” OR “finite element method” OR “frame” AND “Delaunay triangulation” AND “mesh” OR “grid” AND “Generation” AND “canvas” AND “mouse event” NOT “time” NOT “fluid” NOT “Multiscale” NOT “multiscale” NOT “thermal” NOT “dynamic” AND “educational” OR “education” AND “framework”</i>
Periódicos CAPES	<i>“computer graphics” AND “structure” AND “engineering” AND “FEM” OR “finite element method” OR “frame” AND “pré-processador” AND “Delaunay triangulation” AND “mesh” OR “grid” AND “Generation” AND “canvas” AND “mouse event” NOT “time” NOT “fluid” NOT “Multiscale” NOT “multiscale” NOT “thermal” NOT “dynamic” AND “educational” OR “education” AND “framework” AND “graphic program” AND “civil engineer”</i>
Scopus	<i>“computer graphics” AND “structure” AND “education” AND “engineering” AND “FEM” OR “finite element method” OR “frame”</i>

Fonte: Autor (2022)

Nesta segunda etapa deste estudo, foi realizado a execução de acordo com *software* StArt. Contudo, a execução é responsável pela identificação dos estudos selecionados nas bases de dados, seleção de todos os trabalhos que foram aceitos, rejeitados e duplicados, extração dos trabalhos selecionados com leitura dos artigos por completos.

Para ser incluídos no critério de seleção, os artigos deveriam estar relacionados ao uso da computação gráfica para plataforma web aplicada a engenharia de estruturas, que utilizasse computação gráfica com MEF, com relação a engenharia de estruturas e computação gráfica, os textos precisavam estar publicados e disponíveis integralmente nas bases científicas buscadas entre 1983 a 2021 com título, resumo e palavras chaves, escrito com coerência e coesão textual, que foram revisados em pares e com limitação de idioma em português e inglês.

Vale ressaltar que os trabalhos que não tivesse de acordo com os critérios de seleção estaria automaticamente rejeitado. Para ser selecionado os trabalhos tinha que estar de acordo com os critérios de aceitação, leitura dos títulos, abstracts dos artigos. Vale salientar que o *software* StArt tem uma função de localizar todos os trabalhos duplicados e automaticamente organizar na pasta de duplicados.

Após a seleção dos trabalhos seguiu-se para a próxima etapa, na qual os estudos para serem aprovados tinham que ser lidos e analisados integralmente, analisado em pares, estar de acordo com os critérios do formulário de extração de dados e formulário de qualidade conforme a figura 01. Os trabalhos que não foram aceitos na leitura integral automaticamente eram rejeitados.

Figura 01: Formulários de extração e qualidade da RSL

Item	Resposta
Houve coleta de dados	Sim
Houve análise de dados	Sim
Metodologia	
Uso de computação gráfica para desenvolvimento de algum aplicação de MEF	Não
Software de análise de dados	Não utilizado

Item	Resposta
O artigo foi escrito com coerência e coesão textual?	Sim
Os métodos ou técnicas foram reportadas de forma objetiva?	Sim
Caso existam aplicações práticas, elas foram descritas com detalhes?	Sim
Utilizado em engenharia de estrutura?	Não
As utilizações de MEF com computação foram explicitamente citadas?	Não

Fonte: Autor (2022)

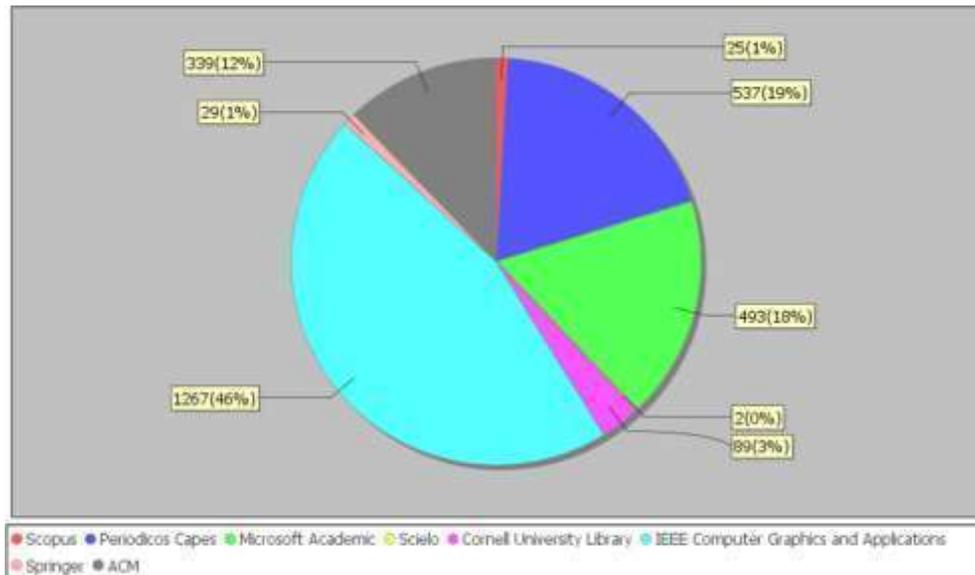
Nesta terceira e última etapa do desenvolvimento desta RSL. A sumarização é listada todos os dados deste estudo em gráficos de pizza e gráfico de barras, na qual apresenta os dados da seleção por bases de dados com total de aceitos, rejeitados e duplicados, assim na extração tendo os mesmos gráficos com dados da extração. Entretanto a sumarização apresenta a nuvem de palavras mais citadas nos trabalhos, gráficos de referências e todos os dados extraídos.

1.3 RESULTADOS E DISCUSSÕES

Para aceitar ou excluir um trabalho existia um formulário de extração e formulário de qualidade que foram planejados no desenvolvimento do protocolo. Estes formulários de qualidade foram necessários para auxiliar a encontrar mais estudos que estão diretamente relacionados com o tema desta revisão sistemática, e, por fim, ser possível averiguar o atual estado da arte.

Dos 2781 trabalhos selecionados para análise, os mesmos foram divididos em 8 bases de dados sendo (12%) ACM, (1%) Springer, (46%) IEEE, (3%) Cornell, (18%) Microsoft Academic, (19%) Periódicos CAPES, (1%) Scopus. Vale salientar que a base de dados Scielo apresentou poucos trabalhos encontrados que não chegou a (1%) dos artigos para análises da revisão sistemática de literatura, conforme a figura 02.

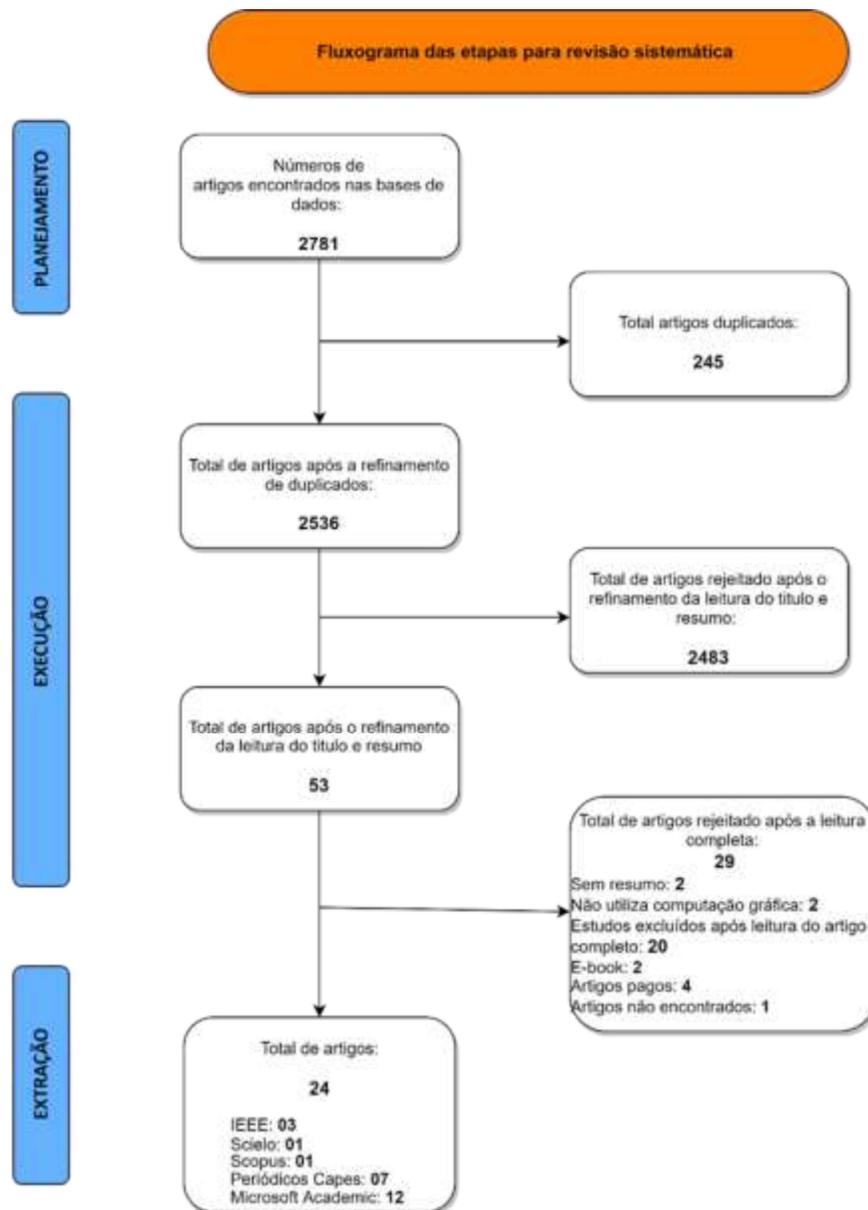
Figura 02: Bases de dados



Fonte: Autor (2022)

A Figura 03 mostra o fluxograma das três fases usadas para encontrar os artigos a serem usados nesta revisão. Na primeira fase, fase de planejamento, foram encontrados um total de 2781 estudos. Na fase de execução, foram excluídos 245 artigos duplicados, 2483 artigos com resumo e *abstract* não correlacionados com o tema desta revisão sistemática. Isso resultou em 53 artigos para leitura completa. Destes, 29 foram rejeitados depois da leitura completa de cada artigo, e o motivo da exclusão foi não apresentar correlação com o tema desta revisão, mas que apresentava potencial quando se leu o resumo. Dessa forma, na fase de extração restaram apenas 24 trabalhos, conforme Figura 03.

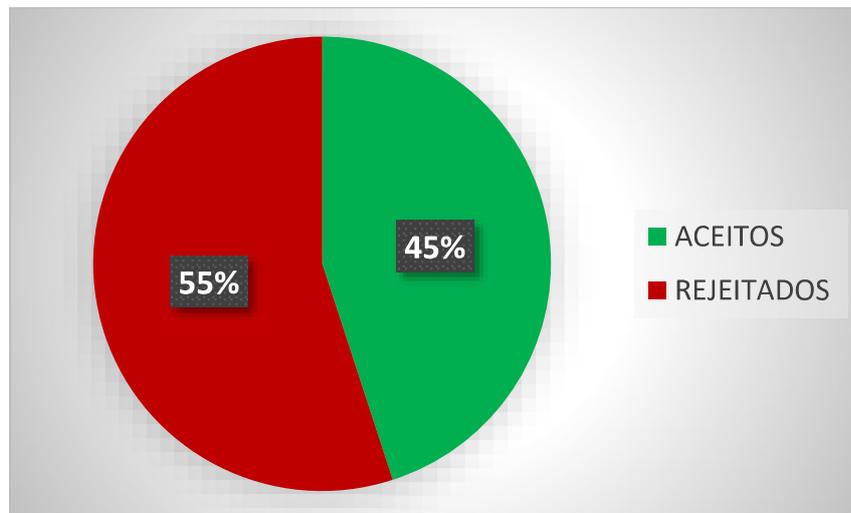
Figura 03: Fluxograma das etapas da RSL



Fonte: Autor (2022)

Os estudos aprovados na extração (55%) foram rejeitados por algum dos motivos que foi planejado no protocolo, sendo eles falta de abstract, artigos disponíveis somente pago e-book e estudos que não correlacionava diretamente com a linha de pesquisa. Contudo (45%) dos trabalhos da extração que foram aceitos porque correlacionava com a linha de pesquisa e utilizava software relacionado aos métodos dos elementos finitos, conforme a figura 04.

Figura 04: Gráfico dos dados da extração



Fonte: Autor (2022)

Para os trabalhos aprovados na extração da revisão sistemática conforme a figura 03 e figura 04, foram relatado uma breve discussão sobre os trabalhos aprovado, os mesmos estão disponíveis em uma lista no apêndice com ano de publicação, revistas publicadas, título e autores.

Com a revisão sistemática de literatura foi possível identificar a evolução dos *softwares* no decorrer da história dos métodos dos elementos finitos, os mesmos surgiram com a criação dos grandes computadores por volta 1980, logo após foram sendo desenvolvidos para aplicações desktops já no período de 1985. Contudo, esses *softwares* tinham um custo muito alto na época, processamento lento e aplicação que somente um usuário utilizava por vez, entretanto os mesmos já geravam malhas bidimensionais e tridimensionais.

A partir do ano de 2005 já é possível identificar estudos de *softwares* web para desenvolvimento de métodos dos elementos finitos, no entanto tinham que utilizar toolkit para OpenGL que possa converter para web, em seguida o Mozilla Foundation desenvolveu WebGL que já não era necessário utilizar o toolkit para converter para web, apesar do contexto histórico de MEF com *softwares* não foi encontrado um software feito com linguagem de programação web em Javascript.

Os trabalhos selecionados na extração apresenta uma determinada variedade de estudos e desenvolvimentos no campo da computação gráfica e análise estrutural, destacando diferentes abordagens e aplicações. Cada trabalho aborda questões específicas, desde o desenvolvimento de sistemas pré e pós-processadores para análise de estruturas planas até a criação de interfaces

interativas para softwares de análise estrutural utilizando técnicas de computação gráfica. Além disso, há discussões sobre o ensino de computação gráfica em diversas áreas do conhecimento, destacando a importância dessa disciplina na formação acadêmica.

A maioria dos artigos mencionados tem relevância para o ensino de métodos de elementos finitos (MEF) e tópicos relacionados a computação gráfica e análise estrutural. Eles abordam uma variedade de tópicos que podem ser úteis para estudantes e profissionais interessados em aprimorar suas habilidades nessas áreas. No entanto, a relevância específica de cada artigo depende dos objetivos e interesses do público-alvo.

Artigos como "*Numerical fundamentals and interactive computer graphics system for the nonlinear analysis of planar frames*" e "*An interactive user interface for a structural analysis software using computer graphics techniques in MATLAB*" são particularmente relevantes para o ensino de MEF, pois tratam do desenvolvimento de ferramentas interativas para análise estrutural, usando software como MATLAB, o que pode ser valioso para estudantes e profissionais que trabalham com análise de estruturas.

Outros artigos, como "*An Efficient Automatic Mesh Generator With Parabolic Arcs in Julia for Computation of TE and TM Modes for Waveguides*" e "*Mesh Simplification Algorithm Based on the Quadratic Error Metric and Triangle Collapse*", exploram aspectos técnicos relacionados à geração de malhas, o que é fundamental em análises de elementos finitos e simulações. Esses artigos podem ser relevantes para alunos que desejam aprofundar seus conhecimentos na área de geração de malhas.

Além disso, artigos relacionados ao ensino de computação gráfica, como "*Distinctive Approaches to Computer Graphics Education*" e "*Teaching introductory computer graphics using java 3D, games and customized software*", podem ser úteis para estudantes que buscam aprender sobre os princípios e aplicações da computação gráfica, embora possam não estar diretamente ligados ao MEF.

Entretanto os artigos mais antigos, como a tradução de software de Pascal para Fortran, destacam a evolução da computação gráfica ao longo do tempo e como as tecnologias foram adaptadas para se adequarem aos avanços nos computadores pessoais. Os trabalhos mencionados podem ser relevantes para o ensino de MEF, mas a escolha de quais deles são mais apropriados dependerá dos objetivos e das necessidades específicas do público-alvo.

Diversos projetos também abordam a integração de tecnologias, como a combinação de scan-to-bim em aplicativos BIM, mostrando a interdisciplinaridade dessas áreas. Além disso, há

uma ênfase no ensino e na criação de ambientes simulados para o aprendizado, como no caso de simulações para a dispersão de medicamentos e o desenvolvimento de plataformas educacionais em computação gráfica.

A maioria dos trabalhos listados apresenta relevância para o ensino de métodos de elementos finitos (MEF) e computação gráfica, especialmente quando aplicados em um ambiente web. Eles demonstram várias abordagens para o uso de tecnologia e software na educação e treinamento em engenharia e disciplinas relacionadas. Aqui estão alguns pontos destacados sobre cada trabalho.

Em resumo, a maioria dos trabalhos apresentados tem relevância no contexto do ensino de métodos de elementos finitos, computação gráfica e disciplinas relacionadas. Alguns deles exploram o potencial de aplicativos web e softwares educacionais para melhorar o aprendizado e a compreensão dos conceitos.

No entanto, a relevância específica para o seu programa de ensino dependerá de seus objetivos e necessidades educacionais, além disto os artigos fornece uma visão abrangente das diversas aplicações, inovações e desafios enfrentados na interseção entre computação gráfica, métodos dos elementos finitos e análise estrutural.

Dos trabalhos aprovados na extração é possível identificar o estudo (MCDOWELL, 2016), que não se trata de MEF e nem computação gráfica. Contudo, o mesmo é uma plataforma digital para ensino, porém para outra área de conhecimento, a aplicação educacional com intuito de agilizar o ensino de farmacêuticos chegou a ganhar dois prêmios internacional.

Portanto, a presente revisão sistemática deixa claro que os estudos sobre MEF e computação gráfica estão relacionados diretamente com diversos tipos de sistemas, porém de acordo com a necessidade, sendo com análise de fraturas, odontologia, medicina e engenharias de estruturas. No entanto, há poucos *softwares* com código fonte aberto para o ensino de MEF nas engenharias, ainda assim os *softwares* que são pagos têm um custo alto e são aplicação desktop e não aplicação web.

Além das aplicações unitários que tem como objetivo de resolução das mesmas tarefas, tem os *softwares* que é possível ter múltiplos soluções na mesma aplicação e precisamente no endereço da web, que foi o estudo do (SONGHUA, 2014). No entanto, essas aplicações requerem um nível de segurança muito eficiente e servidores de alta velocidade, onde foi o empecilho do trabalho citado anteriormente, onde os *firewalls* estava bloqueando o acesso há uma determinada parte do sistema.

1.4 CONCLUSÃO/CONSIDERAÇÕES FINAIS

Apesar dos diversos estudos com desenvolvimento de *softwares* educacional para ensino de métodos dos elementos finitos, é preciso salientar que boa parte desses *softwares* são desktop, deixando o usuário final preso em uma máquina, as poucas aplicações web apresentadas em estudos não estão disponíveis, e as que estão disponíveis há necessidades de atualização.

Foram realizadas buscas em oito bases de dados contendo artigos publicados em vários países com idioma inglês e português no período de 1983 a 2021, os trabalhos aprovados na extração passaram pela análise dos critérios de aceitação e qualidade. Foram utilizado oito *Strings* de busca, no entanto se adequando com cada base de dados, desta maneira foi selecionado 2781 para análise de critérios aceitação, exclusão e qualidade conforme o protocolo que foi planejado antes do início da seleção. Com os tais critérios foi possível chegar a vinte e quatro estudos aprovados na revisão sistemática.

Contudo, com os 24 trabalhos aprovados, boa parte teve desenvolvimento de uma aplicação, seja ela para auxílio na educação, ferramenta de pesquisa ou método de ensino baseado em algum software já existente, possibilitando análise das vantagens e desvantagens de cada tipo de aplicação, sendo elas desktops que são desde da época da criação dos computadores e web. Contudo, um dos empecilhos mais encontrados nos estudos foi a segurança das informações, velocidade de processamento dos servidores ou desktops, interface de difícil manuseio, a falta de treinamento antes do uso do *software* ou manual de uso.

Por fim, esta revisão sistemática de literatura mostrou que o uso de *softwares* para métodos dos elementos finitos tem sido feito em diversas áreas de conhecimento, assim como ferramenta para auxílio nas aulas de engenharia de estruturas nos cursos de graduação e pós-graduação, possibilitando o ensino e aprendizagem mais interativa por meios de interfaces gráficas. Entretanto, buscamos entender o estado da arte sobre o uso de computação com métodos dos elementos finitos, mapeando quais modelos de *softwares* foram mais utilizados e quais eram os mais empecilhos que surgiam. Para trabalhos futuros tende-se no desenvolvimento inicial de uma aplicação web para ensino de métodos dos elementos finitos, que seja utilizada e disponível de acesso gratuito.

REFERÊNCIAS

- ANGEL, Ed. The case for teaching computer graphics with WebGL: A 25-year perspective. **IEEE Computer Graphics and Applications**, v. 37, n. 2, p. 106-112, 2017.
- ANGEL, Ed; SHREINER, Dave. Teaching a shader-based introduction to computer graphics. **IEEE Computer Graphics and Applications**, v. 31, n. 2, p. 9-13, 2011.
- BANFI, F. The integration of a scan-To-hbim process in bim application: The development of an add-in to guide users in autodesk revit. 2019.
- BARROS, M. M. B. **Metodologia Para Implantação de Tecnologias Construtivas Racionalizadas na Produção de Edifícios**. São Paulo, 1996. 410 f. Tese (Doutorado em Engenharia Civil) - Escola Politécnica, Universidade de São Paulo, São Paulo, 1996.
- BLEDZIK, A. K. *et al.* A Comparison of Compounding Process and Wood Type For Wood Fibre-pp Composites. **Composites Part A: applied Science and Manufacturing**, v. 36, p. 789-797, 2005.
- BRANDÃO, D. Q.; HEINECK, L. F. M. Classificação das Formas de Aplicação da Flexibilidade Arquitetônica Planejada em Projetos de Edifícios Residenciais. In: ENCONTRO NACIONAL DE TECNOLOGIA DO AMBIENTE CONSTRUÍDO, 7., Florianópolis, 1998. Anais... Florianópolis: UFSC, ANTAC, 1998. v. 2, p. 215-222.
- C. LOPES, Pedro; L. RANGEL, Rafael; MARTHA, Luiz F. An interactive user interface for a structural analysis software using computer graphics techniques in MATLAB. **Computer Applications in Engineering Education**, v. 29, n. 6, p. 1505-1525, 2021.
- CARDOSO, Maria Clara Santos do Amaral et al. **Dificuldades para o uso da informática no ensino: percepção de professores de matemática após 40 anos da inserção digital no contexto educacional Brasileiro**. Educação Matemática Pesquisa, 2018.
- CHEN, Bo; CHENG, Harry H. Interpretive OpenGL for computer graphics. **Computers & Graphics**, v. 29, n. 3, p. 331-339, 2005.
- COLYER, B.; TROWBRIDGE, C. W. Finite element analysis using a single user computer. **Computer-aided design**, v. 17, n. 3, p. 142-148, 1985.
- COSTA, Fabricio Carneiro; DE SOUZA, Isaac Teixeira; NETO, José de Caldas Simões. A utilização das tic's digitais nos processos de ensino e aprendizagem: uma revisão sistemática. **Brazilian Journal of Development**, v. 5, n. 6, p. 4458-4467, 2019.
- DA SILVA NETO, Sebastião Rogério et al. **Jogos educacionais como ferramenta de auxílio em sala de aula**. In: Anais do Workshop de Informática na Escola. 2013. p. 130-139.
- DE OLIVEIRA TESSARO, Franciele et al. A Tecnologia da Informação na Educação: uma Revisão Bibliográfica. **XVIII Mostra de Iniciação Científica, Pós-graduação, Pesquisa e Extensão**, 2018.
- DERMEVAL, Diego; COELHO, Jorge AP de M.; BITTENCOURT, Ig Ibert. Mapeamento sistemático e revisão sistemática da literatura em informática na educação. JAQUES, Patrícia Augustin; PIMENTEL, Mariano; SIQUEIRA; Sean; BITTENCOURT, Ig(Org.) **Metodologia de Pesquisa em Informática na Educação: Abordagem Quantitativa de Pesquisa**. Porto Alegre: SBC, 2019.
- FRANCOIS, Vincent; CUILIERE, Jean Christophe; GUEURY, Michel. Automatic meshing and remeshing in the simultaneous engineering context. **Research in Engineering Design**, v. 11, n. 1, p. 55-66, 1999.

- GOMES, Gilberto. HEDCPP: Data Structure for Application Programs Aimed to Engineering Education-Learning, 2013.
- ISLOMOVNA, Mamurova Feruza et al. DESIGNING THE METHODOICAL SYSTEM OF THE TEACHING PROCESS OF COMPUTER GRAPHICS FOR THE SPECIALTY OF ENGINEER-BUILDER. **Journal of Contemporary Issues in Business & Government**, v. 27, n. 4, 2021.
- KANNAN, G. Padmasudha; NAGARAJA, K. V. An Efficient Automatic Mesh Generator With Parabolic Arcs in Julia for Computation of TE and TM Modes for Waveguides. **IEEE Access**, v. 8, p. 109508-109521, 2020.
- LAVELLE, J. A. Acrylic Latex-Modified Portland Cement. **ACI Materials Journal**, Detroit, v. 6, n. 1, p. 41-48, jan/feb. 1988.
- LIU, Huan; ZHANG, Hui. A suggestive interface for untangling mathematical knots. **IEEE Transactions on Visualization and Computer Graphics**, v. 27, n. 2, p. 593-602, 2020.
- MA, Songhua; TIAN, Ling. A web service-based multi-disciplinary collaborative simulation platform for complicated product development. **The International Journal of Advanced Manufacturing Technology**, v. 73, n. 5, p. 1033-1047, 2014.
- MARCELINO, Anderson Roman. **A inserção de recursos computacionais na prática pedagógica de professores do ensino técnico e as implicações destas ferramentas na qualificação das aprendizagens**. Anais do II Seminário de Educação Profissional, 2017.
- MARTÍ, Enric; GIL, Debora; JULIÀ, Carme. A PBL experience in the teaching of computer graphics. In: **Computer Graphics Forum**. 9600 Garsington Road, Oxford, OX4 2DQ, UK.: Blackwell Publishing Ltd., 2006. p. 95-103.
- MCDOWELL, Jenny et al. A simulated learning environment for teaching medicine dispensing skills. **American journal of pharmaceutical education**, v. 80, n. 1, 2016.
- OLIVEIRA, Andréa Fernandes de. Gamificação no cotidiano escolar: Um mapeamento sistemático de literatura com ênfase em tecnologia e educação. 2018.
- Oliveira, Jadson Firmo. Desenvolvimento De Uma Aplicação Web Para Gestão Dos Artigos Sobre Aprendizagem Ativa Em Engenharia Publicados No Cobenge, 2018.
- PASTORIO, Dioni Paulo; ALVES, Josemar; FRAGOSO, Tainá Almeida. Uma revisão bibliográfica sobre o uso de software de análise de dados: um olhar para o Octave. **Revista de Ensino de Ciências e Matemática**, v. 12, n. 3, p. 1-23, 2021.
- PETERNIER, Achille; THALMANN, Daniel; VEXO, Frédéric. Mental vision: a computer graphics teaching platform. In: **International Conference on Technologies for E-Learning and Digital Entertainment**. Springer, Berlin, Heidelberg, 2006. p. 223-232.
- RAPOPORT, A. Environmental Cognition in Cross-Cultural Perspective. In: MOORE, G. T.; GOLLEDGE, R. G. (Eds.). **Environmental Knowing**. Stroudsburg: Dowden, Hutchinson and Ross, 1976. p. 220-234.
- RIGGIO, Mariapaola; CHENG, Nancy Yen-wen. Computation and Learning Partnerships: Lessons from Wood Architecture, Engineering, and Construction Integration. **Education Sciences**, v. 11, n. 3, p. 124, 2021.

- SANTANA, Murillo Vinícius Bento; SILVEIRA, Ricardo Azoubel da Mota. Numerical fundamentals and interactive computer graphics system for the nonlinear analysis of planar frames. **REM-International Engineering Journal**, v. 72, p. 199-207, 2019.
- SANTOS, B. Sousa et al. Distinctive approaches to computer graphics education. In: **Computer Graphics Forum**. 2018. p. 403-412.
- SCURATI, Giulia Wally; FERRISE, Francesco. Looking Into a Future Which Hopefully Will Not Become Reality: How Computer Graphics Can Impact Our Behavior—A Study of the Potential of VR. **IEEE Computer Graphics and Applications**, v. 40, n. 5, p. 82-88, 2020.
- SLOBODIANIUK, Olena V. PECULIARITIES OF USING THE METHODOLOGY DISTANCE LEARNING OF THE SUBJECT «ENGINEERING AND COMPUTER GRAPHICS» FOR STUDENTS STUDYING BY CORRESPONDENCE. **Information Technologies and Learning Tools**, v. 17, n. 3, 2010.
- SONG, Xueguan et al. Questionnaire-based discussion of finite element multiphysics simulation software in power electronics. **IEEE Transactions on Power Electronics**, v. 33, n. 8, p. 7010-7020, 2017.
- TORI, Romero; BERNARDES JR, João Luiz; NAKAMURA, Ricardo. Teaching introductory computer graphics using java 3D, games and customized software: a Brazilian experience. In: **ACM SIGGRAPH 2006 Educators program**. 2006. p. 12-es.
- VASILKIN, Andrey. Possibilities of applying structural optimization in building structures computer-aided design systems. In: **MATEC Web of Conferences**. EDP Sciences, 2018. p. 03017.
- ZHOU, Guangyou; YUAN, Shangda; LUO, Sumei. Mesh Simplification Algorithm Based on the Quadratic Error Metric and Triangle Collapse. **IEEE Access**, v. 8, p. 196341-196350, 2020.

**CAPÍTULO 2 – ARTIGO 2
(FOLHA DE ROSTO)**

Título do Artigo: Uma prova de conceito de um aplicativo web para análise de pórticos planos

Qualis: A4

ISSN: 2316-9451

Área: Ciência da computação

Nome da revista: Journal of Interdisciplinary Studies on Science and Informatics

MMC: Uma prova de conceito de um aplicativo web para análise de pórticos planos

MMC: A proof of concept of a web application for analyzing flat frames

RESUMO

O uso de softwares educacional web ainda é pouco explorado nas graduações e pós-graduações dos cursos de Engenharia Civil. Principalmente, pelo alto custo para aquisição de softwares complexos e treinamento de docentes para estar aplicado no processo de ensino-aprendizagem. Portanto, esses mesmos softwares são indispensáveis na graduação, onde o mesmo ajuda o estudante desenvolver suas habilidades e conhecimentos. Entretanto, para os cursos de engenharia é de fundamental importância, onde possibilita o discente a realizar simulações para que não haja possíveis acidentes e desperdícios de materiais. Para o desenvolvimento deste trabalho, foi realizada uma revisão sistemática de literatura para compreender o estado da arte de softwares educacionais web, para ensino de métodos dos elementos finitos que seja gratuito. O resultado desta revisão foi utilizado para desenvolvimento de uma aplicação web com interface gráfica de fácil uso para ensino de métodos dos elementos finitos, com objetivo de inserção de um software educacional no curso de pós-graduação de engenharia aplicada e sustentabilidade (PPGEAS), na disciplina de métodos dos elementos finitos no IFGoiano – *Campus* Rio Verde - GO, o mesmo encontra-se disponível na internet com finalidade que outras instituições públicas e privadas possam usar para uso educacional.

Palavras chaves: educação. MEF. engenharias. software educacional. metodologia de ensino.

ABSTRACT

The use of educational web software is still little explored in undergraduate and graduate courses in Civil Engineering. Mainly due to the high cost of acquiring complex software and training teachers to be applied in the teaching-learning process. Therefore, these same software are indispensable in graduation, where it helps the student to develop their skills and knowledge. However, for engineering courses, it is of fundamental importance where it allows the student to carry out simulations so that there are no possible accidents and waste of materials. For the development of this work, a systematic literature review was carried out to understand the state of the art of web educational software for teaching finite element methods that is free. The result of this review was used to develop a web application with an easy-to-use graphical interface for teaching finite element methods, with the aim of inserting an educational software in the graduate course of applied engineering and sustainability (PPGEAS), in discipline of finite element methods at IFGoiano – Campus Rio Verde - GO, it is available on the internet for other public and private institutions to use for educational use.

Keywords: education. MEF. engineering. educational software. teaching methodology.

1 INTRODUÇÃO

No mundo da educação, a tecnologia revolucionou a forma como os alunos aprendem e interagem com o conteúdo dos cursos superiores. Para MOTTA (2021), com a chegada da internet e dos softwares baseado em nuvem, os alunos agora têm acesso a uma vasta gama de recursos educacionais, desde cursos e tutoriais online até plataformas interativas de aprendizado.

O software educacional cada dia que passa é uma ferramenta crucial para estudantes de engenharia, pois fornece uma riqueza de informações e recursos que podem ajudá-los a desenvolver suas habilidades e conhecimentos. No campo da engenharia, manter-se atualizado com as mais recentes tecnologias, teorias e práticas é essencial, e o software educacional oferece uma maneira conveniente de fazer isso.

Um dos principais benefícios do software educacional na engenharia, é a capacidade de simular cenários e problemas do mundo real. Isso permite que os engenheiros testem suas ideias e soluções em um ambiente seguro e controlado, sem o risco de danificar equipamentos ou colocar pessoas em perigo. Ao usar um software educacional, os estudantes podem experimentar diferentes soluções e ver os resultados em tempo real, dando-lhes uma compreensão mais profunda do assunto e ajudando-os a tomar decisões previamente simuladas. De acordo com Borçato (2022).

Isso ajuda os engenheiros a ampliar sua compreensão de diferentes assuntos com as tecnologias e a aprofundar seus conhecimentos em áreas em que podem ter mais limitações. Esse tipo de software também oferece uma experiência de aprendizado envolvente e interativo, tornando mais fácil para os engenheiros absorver novas informações e reter o que aprenderam. Segundo Santos (2020).

Portanto, fica claro a importância dos softwares educacionais para graduação, pós-graduação e para o ensino de método dos elementos finitos, que é uma técnica numérica utilizada para resolver equações diferenciais parciais que modelam fenômenos físicos em engenharia, ciência e matemática aplicada. Ele é amplamente utilizado em diversas áreas, como engenharia civil, aeroespacial, mecânica, elétrica, entre outras.

A importância do método dos elementos finitos, reside no fato de que ele permite a modelagem e a simulação de sistemas complexos na engenharia e ciência, de maneira mais precisa e eficiente do que as técnicas analíticas tradicionais. Por meio do método dos elementos finitos,

é possível construir modelos matemáticos que representam geometrias complexas, materiais não homogêneos e situações com condições de contorno complexas.

Entretanto, com diversos softwares de simulação e desenvolvimento de MEF que tem disponíveis no mercado, custando um valor acima de \$2500 dólares americanos, impossibilitando os cursos de graduação e pós-graduação o uso dos mesmos Schneider (2019). Contudo, com os softwares gratuitos eles não são disponíveis na web.

Os softwares educacionais são ferramentas essenciais para os cursos de engenharia, pois fornece uma maneira conveniente e eficaz de desenvolver suas habilidades, conhecimento e compreensão. Ao incorporar simulações, recursos e caminhos de aprendizado personalizados, o software educacional ajuda os estudantes a se manterem atualizados com as tecnologias.

Portanto, a presente pesquisa tem como objetivo apresentar o desenvolvimento de uma aplicação web com interface gráfica para ensino de métodos dos elementos finitos, projetado para aprimorar a experiência e aprendizado de alunos em todos os lugares. Este software de ponta combina os mais recentes avanços tecnológicos com uma interface amigável, tornando mais fácil para os alunos acessar e se envolver com o conteúdo dos cursos de graduação e pós-graduação.

2 REFERENCIAL TEÓRICO

Para que seja possível constatar a viabilidade de um software educacional web para ensino de métodos dos elementos finitos, faz-se necessário o estudo de alguns conceitos, tais como, softwares educacionais, programação orientada a objeto, programação *web* e *Javascript*.

2.1 SOFTWARE EDUCACIONAL

Os softwares educacionais são ferramentas importantes no processo de ensino e aprendizagem em cursos de graduação e pós-graduação, pois permitem aos alunos explorar conceitos teóricos em ambientes práticos e interativos segundo Kamanski (2020). Além disso, esses softwares podem ser utilizados para ensinar conceitos avançados e práticas em áreas específicas, bem como para desenvolver habilidades e competências necessárias para a formação do aluno.

De acordo com Leite (2019), algumas das vantagens dos softwares educacionais incluem o acesso a recursos de alta tecnologia: muitos softwares educacionais permitem que os alunos experimentem tecnologias avançadas, como simulações de sistemas complexos, visualizações

3D, realidade virtual e aumentada, que seriam difíceis ou impossíveis de serem criadas manualmente.

Os softwares educacionais permitem que os instrutores personalizem as experiências de aprendizagens de acordo com as necessidades e habilidades individuais dos alunos, permitindo que eles trabalhem em seu próprio ritmo e recebam feedback em tempo real.

Para Fantinati (2021), com desenvolvimento de habilidades práticas os softwares educacionais podem ajudar os alunos a desenvolver habilidades práticas, tais como resolução de problemas, pensamento crítico, análise de dados, programação e design. Para o mercado de trabalho muitos softwares educacionais são usados em empresas e organizações, e a familiaridade com esses programas pode ajudar os alunos a se preparar para o mercado de trabalho e destacar-se em suas carreiras de acordo com Dos Santos (2020).

Entretanto, os softwares que são necessários na preparação do acadêmico para o mercado de trabalho, têm um custo elevado para aquisição para universidades e institutos de educação pública, inviabilizando a aquisição do mesmo e deixando a desejar no currículo do profissional. Contudo, o profissional após sair do curso de graduação sempre procura curso de formação em determinados softwares.

Em resumo, os softwares educacionais são importantes para os cursos de graduação e pós-graduação, pois permitem uma abordagem prática, personalizada e tecnológica para a aprendizagem e ajudam a desenvolver habilidades e competências relevantes para o mercado de trabalho.

2.2 SOFTWARE EDUCACIONAL PARA ANÁLISES DE ESTRUTURAS

Existem diversas aplicações de software na engenharia civil, tanto para cálculos complexos quanto para gerenciamento de projetos e construções. Alguns exemplos de software comumente usados na engenharia civil incluem:

AutoCAD e outros programas de modelagem 3D: esses programas permitem que os engenheiros criem modelos digitais detalhados de estruturas e edifícios, facilitando a visualização e o planejamento do projeto.

SAP2000, ETABS e outros softwares de análise estrutural: esses programas realizam cálculos complexos de análise estrutural, permitindo que os engenheiros verifiquem a segurança e estabilidade de projetos antes de serem construídos.

Software de simulação: programas de simulação, como o *Ansys*, podem ser usados para simular o comportamento de materiais e estruturas sob diferentes condições, permitindo que os engenheiros testem soluções antes de construir.

LUSAS é um software de análise de elementos finitos usado para análise de estruturas civis e mecânicas. Ele suporta uma ampla gama de análises, incluindo análise linear e não linear.

Desenvolvido pela Bentley Systems, o STAAD.Pro é um software de análise e projeto de estruturas que é amplamente utilizado na engenharia civil. Ele suporta análise linear e não linear de estruturas.

Softwares de gerenciamento de projetos: ferramentas como o Microsoft Project ajudam a gerenciar o cronograma, orçamento e recursos de um projeto, permitindo que os engenheiros monitorem o progresso e identifiquem possíveis problemas com antecedência.

Esses são apenas alguns exemplos de como o software é usado na engenharia civil. De acordo com SILVA (2020), com o avanço da tecnologia, novas ferramentas e softwares estão sendo desenvolvidos continuamente para melhorar a eficiência e precisão dos projetos e construções.

Contudo, os softwares citados tem um custo de aquisição muito elevado, custando de 13 mil reais até 13 mil dólares americanos, custo anual da licença do *softwares*. Sendo que apenas o software *Ansys* possui uma licença grátis para estudantes, portanto esta licença tem uma série de restrições no programa.

Para NUNES (2023), no desenvolvimento de softwares para engenharia civil são divididos em duas etapas, sendo elas o pré-processador e pós-processador, os quais são termos comumente usados em computação gráfica, especialmente em softwares de simulação numérica e análise estrutural.

O pré-processador é uma etapa anterior à simulação numérica ou análise estrutural, onde é realizada a preparação dos dados para serem processados de acordo com Rocha (2019). Isso envolve a criação de geometrias, aplicação de condições de contorno, definição de propriedades dos materiais e outras informações necessárias para o modelo. O pré-processador é responsável por criar o modelo numérico que será usado na simulação ou análise.

O pós-processador, por outro lado, é a etapa seguinte à simulação ou análise estrutural. Nessa fase, os resultados da simulação são analisados e visualizados em forma de gráficos, tabelas ou

animações de acordo com Sousa (2021). O pós-processador permite que os engenheiros interpretem e visualizem os resultados da análise, identifiquem possíveis problemas e tomem decisões sobre possíveis mudanças no projeto.

Em resumo, o pré-processador é responsável por criar o modelo numérico, enquanto o pós-processador é responsável por analisar e interpretar os resultados da simulação. Ambas as etapas são essenciais para a realização de análises numéricas e simulações em engenharia.

2.3 PROGRAMAÇÃO ORIENTADA A OBJETOS (POO)

Programação orientada a objetos (POO) é um paradigma de programação que se concentra na representação de objetos do mundo real em forma de software de acordo com Barros (2019). Ele permite a criação de objetos que possuem atributos (dados) e métodos (ações), que podem ser utilizados para modelar e solucionar problemas de maneira mais eficiente.

Em POO, os objetos são criados a partir de uma classe, que serve como um modelo ou "molde" para a criação de objetos. Cada objeto criado a partir de uma classe possui seus próprios valores para os atributos da classe e pode executar seus próprios métodos.

Alguns dos principais conceitos em POO incluem encapsulamento, herança e polimorfismo. O encapsulamento é a ideia de que o estado interno de um objeto deve ser protegido e acessado apenas através de métodos públicos para Sousa (2022). A herança permite que uma classe herde atributos e métodos de outra classe, permitindo a reutilização de código. De acordo com Troquete (2019), o polimorfismo permite que objetos de diferentes classes sejam tratados da mesma maneira, permitindo a criação de interfaces mais flexíveis.

Algumas das linguagens de programação mais populares que utilizam POO incluem Java, Python, C++, C#, Javascript e Ruby. A programação orientada a objetos é amplamente utilizada no desenvolvimento de software, desde aplicativos de desktop e web até jogos e sistemas de gerenciamento de banco de dados.

2.4 METODOS DOS ELEMENTOS FINITOS

O método dos elementos finitos (MEF) é uma técnica numérica utilizada para resolver problemas de engenharia, especialmente aqueles relacionados com a análise de estruturas. Para MESQUITA (2021), o MEF consiste em dividir o domínio do problema em uma série de elementos finitos, onde as equações diferenciais que descrevem o comportamento da estrutura

são resolvidas em cada um desses elementos. Em seguida, as soluções são combinadas para obter uma solução global do problema.

Existem vários softwares educacionais disponíveis para a aprendizagem do método dos elementos finitos, alguns exemplos são:

FreeFEM++: o FreeFEM++ é um software livre e de código aberto que permite a resolução de problemas em várias áreas da física e engenharia, incluindo a análise de estruturas.

Ftool: O Ftool é um software brasileiro amplamente utilizado para análise de estruturas. Ele é adequado para análises de vigas, treliças e estruturas em geral. É uma opção poderosa e de código aberto.

FEniCS: o FEniCS é um software de simulação numérica que utiliza o MEF para resolver problemas em diversas áreas da física e engenharia, incluindo a análise de estruturas.

Calculix: O Calculix é um programa de análise por elementos finitos (FEA) de código aberto. Ele é usado para análise estrutural avançada e oferece recursos poderosos.

SimScale: o SimScale é um software comercial de simulação numérica baseado em nuvem, que oferece recursos avançados para análises de estruturas utilizando o MEF.

OpenSees: O OpenSees (Open System for Earthquake Engineering Simulation) é um software de análise estrutural de código aberto que é amplamente usado em análises de terremotos e simulações sísmicas.

FreeCAD: Embora seja principalmente um software de CAD (Computer-Aided Design), o FreeCAD possui módulos de simulação e análise estrutural. Pode ser usado para análises simples de estruturas.

IDEA Statica Connection: Essa é uma ferramenta gratuita para análise de ligações de aço. É especialmente útil para engenheiros civis que trabalham com estruturas metálicas.

COMSOL Multiphysics: o COMSOL Multiphysics é um software comercial de simulação numérica que inclui módulos específicos para análise de estruturas utilizando o MEF.

Xtructure: Um software de análise estrutural simples e de código aberto que se concentra em análises de vigas e treliças.

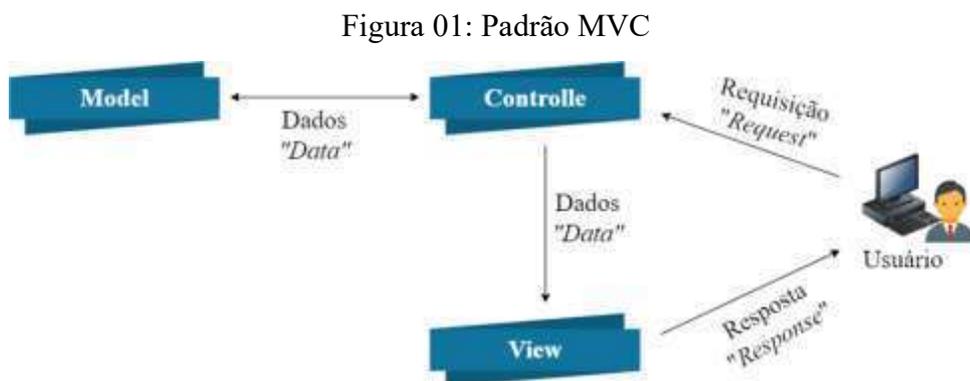
Esses softwares oferecem recursos para modelagem, análise e pós-processamento de problemas de engenharia, permitindo que estudantes e pesquisadores aprendam e apliquem o método dos elementos finitos em suas pesquisas e trabalhos. Além disso, existem diversos materiais educacionais, como tutoriais e livros, que abordam o método dos elementos finitos e seu uso em problemas de engenharia.

3 MÉTODO

A metodologia aplicada neste projeto é a pesquisa experimental tecnológica, com objetivo no desenvolvimento do software MEFWEB. Seguindo as recomendações da NBR 6118 (2014), contemplando diversos módulos da mesma tais como suporte, restrições e análise de estruturas de nós fixos.

Diante disto, o software foi desenvolvido na linguagem de programação Javascript, por ser uma linguagem mais documentada no momento. Vale ressaltar que a mesma é web e programação orientada a objetos (POO). Com o objetivo de desenvolver do mesmo foi escolhido um ambiente de desenvolvimento integrado (IDE) sendo o *Visual Studio Code* (VSCode) que é uma ferramenta gratuita com diversos plugins.

Para o desenvolvimento do MEFWEB foi utilizado o padrão *Model-View-Controller* (MVC), o mesmo visa a padronização do código fonte, possibilitando que futuros programadores que possam ingressar no projeto seja capaz de compreender todo código fonte da aplicação sem a necessidade do seu criador. O padrão MVC ele é dividido em três etapas sendo model, view e controlle conforme a figura 01 abaixo.



Fonte: Autor (2022)

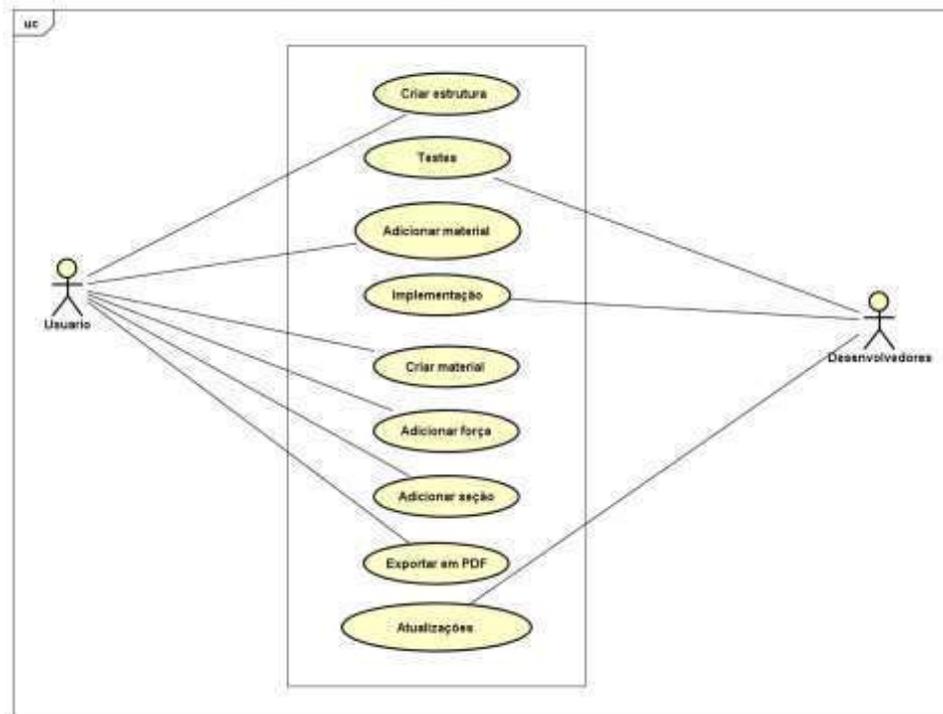
Conforme a padronização MVC da figura 01, o desenvolvimento da aplicação ocorreu em três etapas, sendo model organizando e padronizando a manipulação de dados, view sendo criação da interface gráfica, controlle criando todas as requisições do sistema. Diante disto, foi

desenvolvido os diagramas de caso de uso para descrever as funcionalidades e propostas do sistema antes do seu desenvolvimento, com intuito de levantamento de requisitos da aplicação.

O diagrama caso de uso foi desenvolvido com dois atores, sendo eles o desenvolvedor com intuito de realizar as atualizações, implementação e testes da aplicação. Por outro lado, temos o usuário que tem diversas possibilidades desde criar estruturas, adicionar material, criar material, adicionar força, adicionar seção transversal e exportar sua determinada estrutura em PDF na versão disponível conforme a figura 02. Vale salientar que nas atualizações futuras terá outros tipos de dados a ser exportado, desde imagem ou até mesmo a edição do próprio arquivo exportado.

O caso de uso tem seu papel fundamental no desenvolvimento da aplicação, tendo em vista que o mesmo é responsável por descrever os casos de uso na fase inicial do projeto, possibilitando que o desenvolvedor possa iniciar todo desenvolvimento da aplicação, seja ela em web ou desktop.

Figura 02: Diagrama de caso de uso

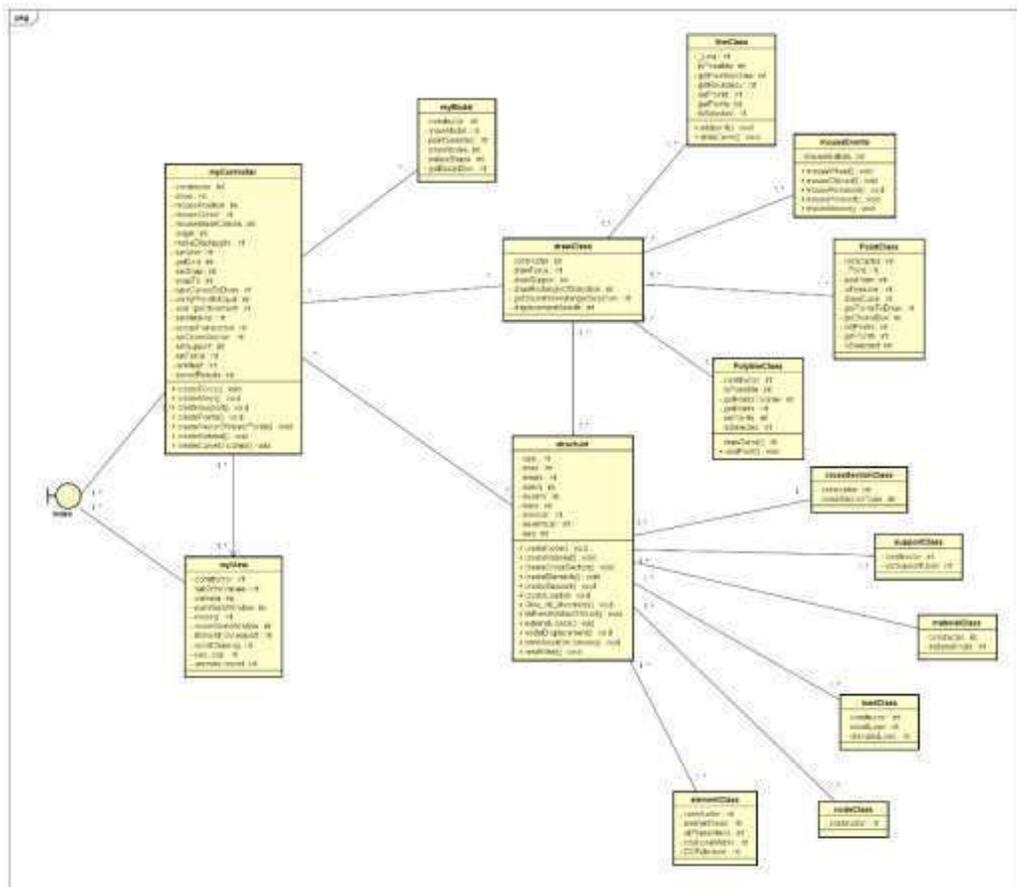


Fonte: Autor (2023)

No entanto, no decorrer do desenvolvimento da aplicação não resultou somente no diagrama de caso de uso, tivemos que desenvolver a representação das estruturas por meios do diagrama de classes. O Diagrama de classes ele é composto pelo atributo e métodos. Os atributos são representados com um traço antes do nome, já métodos são representados com um sinal de mais (+) antes do nome, os parâmetros dos métodos são listados entre parênteses, quanto ao tipo de retorno dos métodos é indicado após os dois pontos (:), conforme a figura 03.

O mesmo foi desenvolvido por desempenhar um papel crucial em todas as fases do desenvolvimento de software orientado a objetos, desde a análise e o design até a implementação, a comunicação e a manutenção do sistema, possibilitando que o desenvolvedor possa identificar classes reutilizáveis e ajuda a identificar as responsabilidades de cada classe.

Figura 03: Diagrama de classes



Fonte: Autor (2023)

No desenvolvimento da modelagem para análise e design das classes, o diagrama de classes foi desenvolvido com as seguintes classes *elementClass*, *nodeClass*, *loadClass*, *materialClass*, *supportClass*, *crossSectionClass*, *structure*, *myView*, *myController*, *myModel*, *drawClass*, *myModel*, *mouseEvents* e *lineClass*, sendo que algumas são classes completa com atributos e métodos. Diante das diversas classes criada podemos citas as cinco classes mais importantes, sendo *structure*, *myView*, *myController*, *myModel*, *drawClass*, que serão detalhadas logo abaixo.

A classe *MyView* é responsável pela comunicação entre o *MyCrontrolle* e index sendo uma das partes mais importantes, pois a mesma é a parte visual da aplicação reproduzindo e desenvolvendo as estruturas. Vale ressaltar que embora a mesma tenha diversos atributos e realiza a reprodução das solicitações do usuário ao *MyCrontrolle*.

O *MyController* é a classe que recebe as solicitações do index e encaminha as mesmas solicitações para as outras classes o mesmo aguarda as demais classes responder e encaminha

os dados para a tela do usuário pela classe o *MyView*. A classe *MyControlle* é composta pelos seguintes atributos *constructor*, *show*, *mousePosition*, *mouseCursor*, *mouseInsideCanvas*, *origin*, *makeDisplaygrid*, *setGrid*, *getGrid*, *setSnap*, *snapTo*, *typeCurvesToDraw*, *verifyIfPointIsEqual*, *addTipeOfElement*, *setMaterial*, *secaoTransversal*, *setCrossSection*, *setSupport*, *setForce*, *runMesh* e *storedResults* nas mesmas é possível identificar que os atributos conforme a função da classe que é alterado por meio da função *set* e solicitado pelo *get*. Além dos atributos, a classe possui os métodos ou operações *createForce*, *createMesh*, *createSupport*, *createCurvesToDraw*, *createVectorOfobjectPoints*, *createMaterial* e *createPoints* os métodos citados acima são responsáveis pela operação, entretanto na classe *controlle* não cria, a mesma solicita por meio do *get* e *set* as outras respectivas classes que estão relacionadas.

A *MyModel* é uma classe que tem sua importância nos sistemas com grandes estruturas de dados, entretanto na aplicação que foi desenvolvida ela possui alguns atributos, tais como *constructor*, *showModel*, *pointSelected*, *showNodes*, *deleteShape* e *getBoundingBox*, ambas classes têm suas manipulação de dados por meio de identificação, se o ponto está selecionado ou não ou por meio do *get* e *set*.

No decorrer do desenvolvimento das classes o *DrawClass* é uma das mais importantes da construção do MEFWEB tendo em vista que a mesma é uma classe de desenho. A mesma possui alguns atributos tal como *constructor*, *drawForce*, *drawSupport*, *drawRectangleOfSelection*, *getBoundingBoxRatangleSelection* e *displacementResults*, embora a classe tenha o *get* e diversos desenho está associada a outras classes, podemos citar a *lineClass* que é responsável pelo desenho da reta no plano, o *mouseEvents* identifica os movimentos do ponteiro do mouse, *PointClass* desenha somente um ponto ou nó no plano e *PolylineClass* tem como objetivo desenhar diversas linhas com nós em uma determinada estrutura.

A classe *Structure* na aplicação é uma classe que tem associação com outras classes, sendo a mais importante do projeto para área de engenharia de estruturas, pois a mesma tem os seguintes atributos conforme a tabela 01:

Tabela 01: Atributos da classe *Structure*

- type :	Determina o tipo da estrutura se é TP1 ou TP2;
- nnos:	Número de nós na estrutura;
- nmats:	Número de quantidade material na estrutura;

- nsecs:	Estabelece o número de seções transversais da estrutura;
- nelems:	Define o número de elementos da estrutura;
- naps:	Estabelece o número de suportes na estrutura;
- nnoscar:	Indica o número de nós carregados na estrutura;
- nelemscar:	Define o número de elementos da estrutura;
- neq:	Estipula o número de equações para resolver o problema;

Fonte: Autor (2023)

Desta maneira a classe *Structure* tem todos os elementos necessários para o desenvolvimento de uma determinada estrutura, tendo em vista que nesta classe são feitas as entradas de dados da estrutura a ser manipulado. Além dos atributos, a mesma classe tem os métodos que é responsável pelas operações, tais como *createNode*, *createMaterial*, *createCrossSection*, *createElements*, *createSupport*, *createLoads*, *Grau_de_liberdade*, *stiffnessMatrixOfStruct*, *externalLoads*, *nodalDisplacement*, *printResultOnConsole* e *resetMest*, os determinados métodos estão associados com outras classes.

O método *createNode* ele cria um novo nó que é adicionado na matriz de coordenadas para compor a estrutura, *createMaterial* cria um novo material que é adicionado na matriz de materiais para compor a estrutura, *createCrossSection* este método é usado para criar uma nova seção transversal que é definido na matriz apropriada.

CreateElements este método é usado para criar novos elementos o mesmo também é definido na matriz apropriada, *createSupport* ele cria novos suportes que são definidos na matriz apropriada, definindo o novo suporte da estrutura, possibilitando a identificação do eixo *rtx* e *rtz* que está impedido ou livre, *createLoads* o presente método é usado para criar novas cargas nodais da estrutura definido pela matriz apropriada, o mesmo identifica os números de nós carregados, *Grau_de_liberdade* para esta versão da aplicação todos os elementos devem ser iguais, o número do grau de liberdade "nglno" é o mesmo para todos os elementos, neste método basicamente o que é feito é pegar da matriz de restrições do apoio do nó restringido e depois esse nó será a linha da matriz glno onde serão adicionados os graus de liberdade da matriz de restrição.

StiffnessMatrixOfStruct o determinado método cria o tamanho do vetor de graus de liberdade do elemento, por exemplo, se $nglno = 2$ (graus de liberdade do elemento), então o número de graus de liberdade por elemento será 4. Assim, este vetor irá guardar, para o exemplo anterior, 4 graus de liberdade, sendo 2 graus por nó, portanto, o vetor $gle = (gdl\ x\ do\ nó\ 1, gdl\ y\ do\ nó$

1, gdl x do nó 2, gdl y do nó 2) totalizando 4 posições, *externalLoads* forças externas nos nós possibilitando a implementação de elementos carregados, *nodalDisplacement* o presente método realiza a avaliação do nodal de deslocamento livre, *printResultOnConsole* o método printa todos os dados da estrutura no console log e o método *resetMest* ele apaga todos os dados que foram inserido possibilitando que o usuário possa adicionar novos dados de uma nova estrutura.

4 RESULTADOS E DISCUSSÕES

Nesta seção será apresentado todos resultados que foi desenvolvido na pesquisa de dissertação, com intuito de divulgar todas as funcionalidades da ferramenta, conforme foi planejado na metodologia e no diagrama de caso de uso para o desenvolvimento do software educacional.

4.1 PÁGINA INICIAL DO SISTEMA MEFWEB

Na interface do site do sistema temos um menu no canto superior direito da tela, que tem os seguintes hiperlinks para a tela de início, sobre, vantagens, argumentos e contatos. Na parte inferior é composto por hiperlinks que interliga as redes sociais tais como youtube, facebook, GitHub e Instagram. A interface está em formato de slidebar que tem um carrossel de imagens, possuindo dois botões para redirecionamento para o sistema MEFWEB, o primeiro botão está na logo do sistema na parte superior da interface e o segundo está destacado na parte central da mesma com nome “conheça mefweb” conforme mostra a figura 01.

De acordo com menu da interface do site temos as seguintes telas, vantagens e sobre. Que tem como objetivo trazer informações sobre o que é o sistema MEFWEB com detalhamento das suas vantagens por ser uma ferramenta educacional, podendo ser utilizado em algumas disciplinas tal como métodos dos elementos finitos. Vale ressaltar que a mesma é uma ferramenta totalmente gratuita, buscando se diferenciar em meios a outras ferramentas que são pagas chegando a impossibilitar que instituições de ensino possa adquirir.

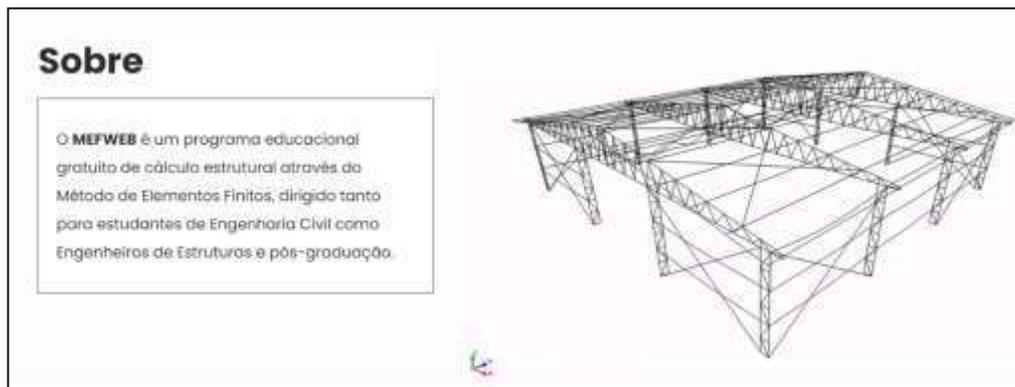
A ferramenta educacional traz uma interface intuitiva de fácil uso, com uma simplicidade para que o usuário tenha uma facilidade no desenvolvimento de suas atividades. É importante salientar que além dos benefícios citados no uso da ferramenta, o site redireciona o usuário para a aplicação que é web e não para um gerenciador de download tendo em vista, que a aplicação é web conforme a figura 02 e 04.

Figura 04: Página inicial



Fonte: Autor (2023)

Figura 05: Página sobre



Fonte: Autor (2023)

Figura 06: Vantagens do MEFWEB



Fonte: Autor (2023)

Ficou disponível no site da ferramenta educacional uma aba para argumentos que tem como objetivo trazer alguns argumentos, de profissionais da educação desde a área da engenharia,

computação e educação, relatando a importância das ferramentas educacionais para auxílio no ensino desde a educação profissional até educação básica. No rodapé do site ficou disponível informações sobre a ferramenta, contatos com endereço, telefone, e-mail e redes sociais com facebook, Instagram, youtube e GitHub.

Figura 07: Argumentos e contatos



Fonte: Autor (2023)

4.2 INTERFACE DO SISTEMA MEFWEB

A tela inicial do software é uma interface intuitiva e fácil de utilizar, sendo dividida em cinco partes, onde a parte superior compõe os botões de centralizar de zoom in e out, mover para esquerda e direita, com semelhança os botões de para cima e baixo. Os botões citados anteriormente são os que controla o centro da interface gráfica do MEFWEB conforme mostra a figura 06. Entretanto, além dos botões citados tem o botão de linha, polyline, ponto, seção transversal, força, malha, rum e download ambos, serão detalhados nos parágrafos seguintes. O botão *polyline* é responsável por desenhar figuras planas, tal como polígonos que é composta por segmentos de reta interligados. Os polígonos mais comuns são triângulos, quadriláteros, pentágonos, hexágonos, entres outros conforme mostra a figura 06. A geometria poligonal é amplamente aplicada em diversas áreas, como na engenharia civil, na arquitetura e computação gráfica. O *line* tem como objetivo desenhar uma reta na interface principal, ligando de um determinado ponto que esteja no eixo x e y para outro ponto no eixo x e y.

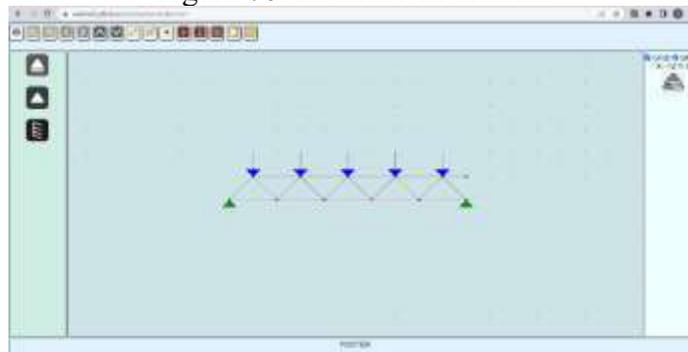
Point é um botão que tem como intuito gerar um ponto em uma coordenada no eixo x e y no canva. Vale ressaltar que no canva é permitido gerar pontos nodais, é possível criar diversos outros pontos nodais e não somente um ponto. *Force* o botão tem como objetivo a inserção de

força sobre uma determinada estrutura seja ela tração ou compressão, tendo em vista que a análise da força é essencial para garantir a segurança e a estabilidade das estruturas e para evitar falhas e colapsos.

O botão seção transversal é usado para analisar as características internas de um tipo de estrutura, ao analisar a seção transversal de uma estrutura, é possível obter informações importantes sobre sua geometria, como sua área, resistência, rigidez e elasticidade, conforme mostra a figura 06. Com intuito a análise de elementos finitos o botão *mesh* tem como objetivo criar modelos de estruturas, por meio de pontos nodais interconectados por linhas em uma superfície plana.

Os últimos botões é o *Run* e download, o *Run* funciona ao ser clicado, a partir disto o software compila o código fonte e executa o programa conforme a força, seção e material adicionado sobre determinada estrutura. O último botão da barra de tarefa superior é o download foi criado com intuito que o usuário pudesse baixar toda sua estrutura em PDF ou JPEG no momento inicial. Entretanto, vale ressaltar que o mesmo vai ter atualização futura em que o usuário possa baixar um material e logo após possa realizar o upload do mesmo para realizar determinadas edição.

Figura 08: Interface do sistema



Fonte: Autor (2023)

No guia menu lado esquerdo de acordo com a figura 06, é possível identificar três botões sendo eles primeiro gênero, segundo gênero e terceiro gênero, ambos são elementos estruturais definidos, sendo conhecidos como vínculos de apoio onde tem como objetivo de impedir o movimento de uma estrutura. Portanto, os mesmos foram definidos por se trabalhar em estruturas planas, contudo o nosso primeiro botão descrente corresponde ao primeiro gênero, sendo responsável por impedir o movimento normal ao plano de apoio. O segundo gênero é o que impede os dois movimentos, seja na horizontal ou vertical, portanto, é possível formar até

duas reações. Contudo, o terceiro gênero tem como objetivo de impedir o movimento de ambos os lados impedindo também a rotação.

Na guia do menu direito é possível identificar dois *check box* sendo o primeiro *grid* e o segundo o *snap*, ambos desseleccionado são desativados, no *grid* é possível notar que os pontos nodais irão sair do centro da tela. Logo abaixo na mesma guia temos a posição do cursor do eixo x e y na tela com a determinada coordenada. No mesmo menu no último campo temos o botão do material, ao ser clicado ele irá abrir um *pop-up* com os três materiais pré-definido deixando o usuário selecionar qual deles desejado, sendo eles aço, concreto e alumínio. Caso o usuário necessita de um que não seja nenhum dos citados, ele pode digitar o valor no campo e logo após clicar no botão *submit* para submeter o material conforme a figura 07.

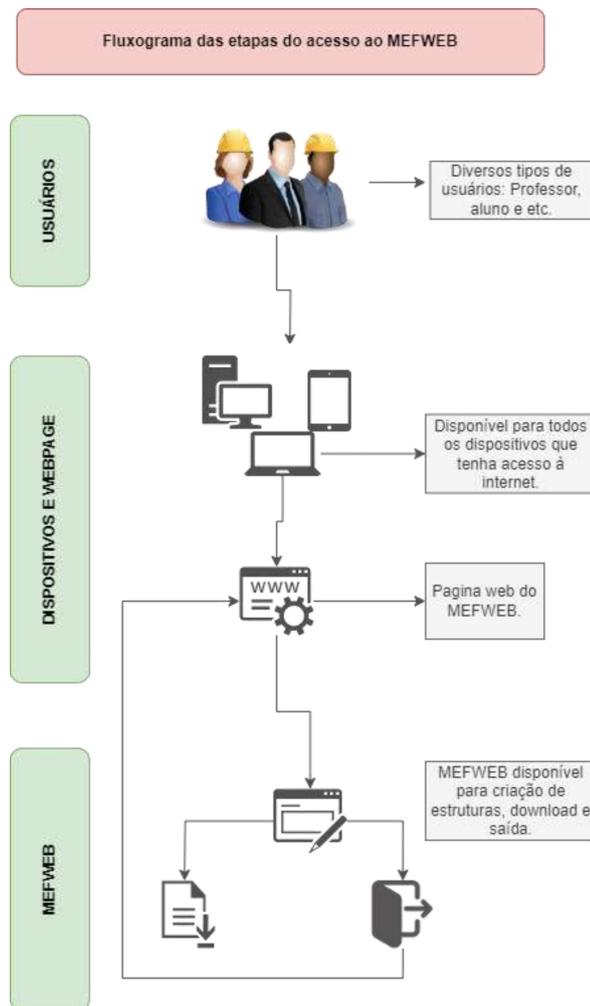
Figura 09: Tipo do material



Fonte: Autor (2023)

Para compreendermos melhor as etapas da aplicação, foi desenvolvido um fluxograma para esclarecer melhor, conforme a figura 09. O fluxograma foi dividido em três etapas, sendo a primeira composta pelos usuários que podem ser alunos, professores e engenheiros. A segunda etapa identifica que MEFWEB pode ser acessado de todos os dispositivos que tenha acesso a internet, desde a *webpage* até a página principal. Terceira e última etapa é a página principal onde o aluno pode desenvolver suas estruturas com métodos dos elementos finitos, na mesma página o mesmo pode realizar o download do material ou sair, ao clicar no botão de sair o usuário será redirecionado para *webpage* que é a página de apresentação da aplicação.

Figura 10: Fluxograma MEFWEB



Fonte: Autor (2023)

Foi desenvolvido testes de aceitação e funcional para identificar se a aplicação funciona corretamente sem nenhuma falha, desta maneira possibilita uma atividade de validação do mesmo. Todos os testes foram desenvolvidos no *selenium* na versão *web* que está disponível como extensão do *google Chrome*. Portanto, na aplicação há diversos tipos de testes que tem que ser realizado, tal como teste de desempenho, carga, estresse, regressivo e orientado a comportamento, embora alguns dos testes citados seja desenvolvido em *software*, vai ser possível identificar alguns bugs quando o usuário estiver usando.

A aplicação foi hospedada gratuitamente no servidor do *GitHub*, no endereço eletrônico <https://webmef.github.io/edu>, com informações sobre os desenvolvedores, tais como nome e formação, este mesmo endereço é a página de entrada para o software. Será disponibilizado tutorial por meio de vídeos e PDF para melhor facilidade da interação do usuário com a plataforma, vale salientar que no endereço eletrônico do *GitHub* <https://github.com/webmef/edu>, está disponível alguns canais de comunicação tais como e-mail,

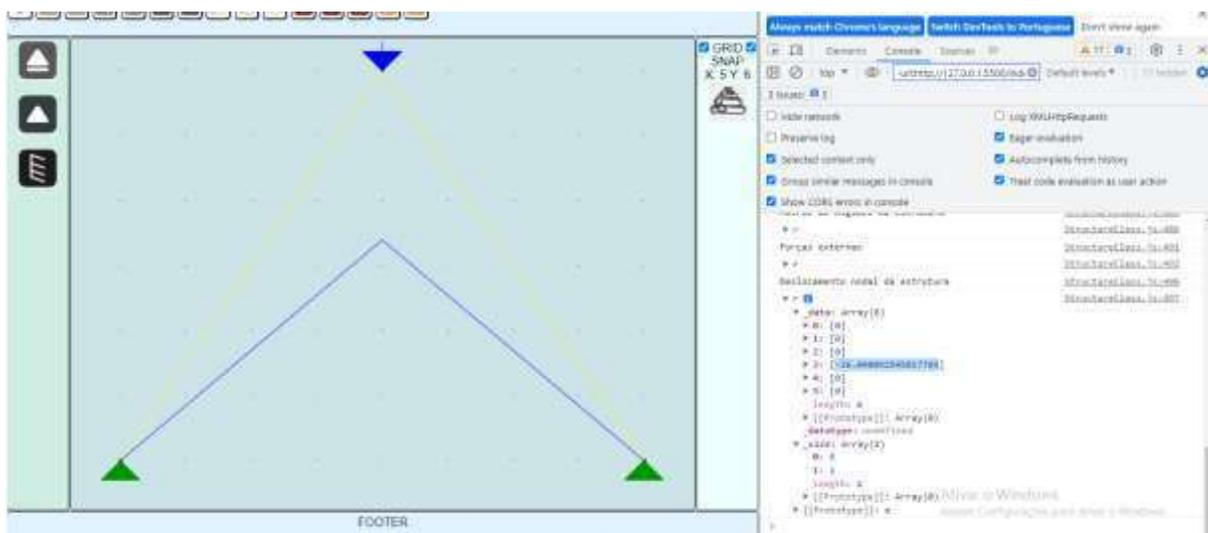
telefone para receber elogios, críticas e sugestões de atualização, além dos canais de comunicação, está disponível as linguagens que a aplicação foi desenvolvida. Vale ressaltar que o código fonte está disponível no mesmo endereço eletrônico do GitHub.

As ferramentas de análise estrutural geralmente possuem interfaces intuitivas que permitem a modelagem e a visualização de estruturas em 2D ou 3D. Além disso, elas fornecem recursos para aplicação de cargas, definição de materiais, solução de equações de equilíbrio e análise dos resultados.

Foi desenvolvido uma estrutura para o resultado desta pesquisa, sendo da aplicação que já está disponível no link: <https://webmef.github.io/edu/> como link principal do software, possibilitando a análise estrutural por meio de interação gráfica do MEFWEB. Entretanto, a estrutura foi criado com três pontos nodais, com dois suporte, seção transversal e uma carga em um determinado nó, conforme mostra a figura 11.

É importante salientar, que a força aplicada no MEFWEB ela já é pré-definida na sua programação, tendo a opção para ser adicionada o material conforme a necessidade do usuário. Vale ressaltar que o MEFWEB está sendo desenvolvido, sendo um modelo para aprimoramento nas atualizações futuras que será disponível as opção para selecionar as seção transversal e os campos da força em concordância com a necessidade do usuário.

Figura 11: Estrutura no MEFWEB



Fonte: Autor (2023)

Após a análise do programa, é possível identificar que o software obteve o deslocamento de 26,04 mm, e este resultado é igual ao deslocamento encontrado pela teoria de resistência dos

matérias para deslocamento de barras. Isso significa que o software apresentou um deslocamento coerente, validando o software para este exemplo. Visto que o MEFWEB já está realizando online e de forma gratuita o desenvolvimento de vários tipos de estruturas garantindo o ensino de métodos dos elementos finitos.

Contudo, embora que MEFWEB não tenha sido usado ainda com grande quantidade de estudantes, é importante salientar que a aplicação é educacional, gratuita e disponível na internet, necessitando que o estudante ou professor precise somente de um dispositivo que tenha acesso à internet e não de grandes computadores com grande capacidade de processamento.

5 CONCLUSÕES

Neste artigo foi desenvolvido um *software* educacional para ensino de métodos dos elementos finitos no Instituto Federal Goiano Campus Rio Verde – GO, para o curso de Engenharia Civil. Com a necessidade de um software para desenvolvimento de métodos dos elementos finitos que seja gratuito, foi possível fazer a integração com diversas áreas de conhecimento tais como computação, ensino, aprendizagem e educação para o desenvolvimento do mesmo.

A ferramenta educacional MEFWEB necessita de internet para o seu uso de forma rápida e eficiente, sendo que a mesma não precisa de um computador com alta capacidade de processamento, tendo em vista que todo seu processamento é no navegador. Desta maneira, possibilitando que várias instituições de ensino possam utilizar sem ter nenhum custo com aquisição de licenças.

Vale salientar que MEFWEB não será utilizado exclusivamente para instituições de ensino públicas, sendo capaz de atender toda comunidade em todos os segmentos, seja privado ou publica com intuito de divulgação. Será realizado uma pesquisa após o uso do MEFWEB na sala de aula, com objetivo de analisar, avaliar e receber sugestões dos professores e acadêmicos de diferentes períodos para que podemos realizar uma atualização conforme a necessidades dos usuários.

O MEFWEB foi desenvolvido pelo aluno e professor do programa de pós-graduação em engenharia aplicada e sustentabilidade (PPGEAS). Para trabalhos futuros será continuado o seu desenvolvimento para melhorias, tais como visualização em 3D, simulações de vigas, treliças e atualização do mesmo para auxiliar o ensino das aulas de métodos dos elementos finitos nos cursos de engenharia e que continue sendo gratuito e web. Espera-se que todo desenvolvimento

deste projeto de pesquisa contribua diretamente nos cursos de engenharia e no ensino e aprendizagem.

REFERÊNCIAS

As referências devem ser reunidas no final do artigo em uma única ordem alfabética, de acordo com NBR 6023 - Referências – Elaboração (2002). Alinhado à esquerda e entrelinhas depois em 6 pontos.

BARROS, Alexandre Monteiro de; SILVA, Regio Pierre da; TEIXEIRA, Fabio Goncalves. Representação do conhecimento de projeto de produto utilizando o paradigma da orientação a objetos. Blucher Design Proceedings. Blucher, 2019.

BORÇATO, Allan Guimarães; Hackbarth, Crizane; Melo, Nicolay Damaceno Aguida. USO DE SOFTWARES NO PROCESSO DE ENSINO-APRENDIZAGEM DOS CURSOS DE ENGENHARIA CIVIL DA EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA. Revista Extensão & Cidadania, v. 10, n. 17, p. 84-100, 2022.

DOS SANTOS, Joaffson Felipe Costa et al. Efeito da modalidade de ensino presencial comparado ao da utilização de software educacional no conhecimento de acadêmicos de Enfermagem. Brazilian Journal of Development, v. 6, n. 3, p. 15010-15024, 2020.

FANTINATI, Regiane Ezequiel; DOS SANTOS ROSA, Selma. Pensamento computacional: Habilidades, estratégias e desafios na educação básica. Informática na educação: teoria & prática, v. 24, n. 1 Jan/Abr, 2021.

KAMANSKI, Angela Maria Baruffi et al. Aplicativos e softwares educacionais: percepções de professores da área de Ciências da Natureza. I Simpósio Sul-Americano de Pesquisa em Ensino de Ciências, n. 1, 2020.

LEITE, Maici; REINALDO, Francisco; OLIVEIRA, Quelli. Ações de intervenção na formação de professores com base em metodologia tecnológica. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. 2019. p. 61.

MESQUITA, Camila et al. Aplicação do método dos elementos finitos na modelagem de concreto poroso com aplicação em pisos permeáveis. 2021.

MOTTA, Edmilson de Sá. Desenvolvimento de uma ferramenta para disponibilização de conteúdo hipermédia adaptativo. 2021.

NUNES, João Fyllipy de Lima et al. Desenvolvimento de software web para análise estrutural não linear geométrica de pórticos plano. 2023.

ROCHA SEGUNDO, Jackson da Silva. Estratégias numéricas para otimização da solução não linear de problemas estruturais. 2019.

SANTOS, Ricardo Tiburcio dos. A engenharia didático-informática: uma metodologia para a produção de software educativo. 2020.

SCHNEIDER, Jorge. Softwares educacionais para o ensino da matemática no ensino médio. 2019.

SILVA, Rafael Soares. As aplicações da informática no ensino de química como alternativa para o ensino e aprendizagem. Revista Amor Mundi, v. 1, n. 2, p. 43-56, 2020.

SOUSA, EMERSON L. et al. Desenvolvimento do pós-processador matemático para o software Free-IAT. 2021.

SOUSA, Mikael Lucas de Brito. Desenvolvimento de sistema de monitoramento de câmeras para detecção de aglomeração e máscaras de proteção individual através de redes neurais. 2022.

TROQUETE, Rafael. Programação orientada a objetos: uma visão conceitual dos elementos de modelagem. 2019.

CONSIDERAÇÕES FINAIS

O presente trabalho teve como intuito o seu desenvolvimento no decorrer da pós-graduação, apresenta como objetivo o desenvolvimento de uma aplicação web para análise e deformação com métodos dos elementos finitos, é *Open Source* onde tem o seu código fonte aberto com finalidade de receber contribuições, tanto internas e externas da comunidade do IFGoiano. Vale ressaltar que o grande diferencial da aplicação para outros softwares é a disponibilidade para o usuário em qualquer localidade com acesso à internet e qualquer horário, e para todas as plataformas disponíveis. É de suma importância lembrar que a aplicação foi um trabalho inicial.

Uma das principais características do software é a sua interface amigável, que permite aos usuários visualizar e manipular modelos estruturais, aplicar condições de contorno, selecionar diferentes tipos de elementos e analisar os resultados de forma interativa. Acreditamos que nosso software web educacional tem o potencial de impactar positivamente o ensino e a aprendizagem de métodos dos elementos finitos. Além de proporcionar uma experiência interativa e envolvente, ele permite que os estudantes adquiram habilidades práticas e apliquem os conhecimentos teóricos em situações reais.

O projeto foi um trabalho inicial, que tem com objetivos trabalhos futuros, tais como implementação de elementos 3D, criação de cadastro de usuários com intuito que cada usuário possa ter seu ambiente de desenvolvimento próprio, atualização para implementação de recursos gráficos, criação da função do apagador, criar um tipo de arquivo que seja possível o usuário realizar upload e edição do mesmo, possibilitar mais opções de download, desenvolver um executável para mais de um sistema operacional para instalação em caso do usuário que não tenham internet, adicionar recursos avançados, como a integração com software de CAD, a incorporação de análises dinâmicas e a ampliação do escopo para abranger diferentes áreas da engenharia.

Em conclusão, o desenvolvimento do MEFWEB para o ensino de métodos dos elementos finitos representa um passo importante na promoção da educação na engenharia.

Esperamos que essa ferramenta possa contribuir para a formação de estudantes mais preparados e capacitados, impulsionando avanços na análise estrutural e promovendo a inovação em diversas áreas da engenharia.

APÊNDICES E ANEXOS

APÊNDICE A - LISTA DE TRABALHOS DA EXTRAÇÃO DA RSL

Ano de Publicação	Revista	Título do Artigo	Autores
1970	Information Technologies and Learning Tools	Peculiarities of using the methodology distance learning of the subject «engineering and computer graphics» for students studying by correspondence	Slobodianiuk, Olena V.
1970	IOSR Journal of Engineering	HEDCPP: Data Structure for Application Programs Aimed to Engineering Education-Learning	Gomes, Gilberto
1985	Computer aided design	Finite element analysis using a single user computer	Colyer, B ; Trowbridge, C.W
1999	Research in engineering design	Automatic Meshing and Remeshing in the Simultaneous Engineering Context	François, Vincent ; Cuillière, Jean Christophe ; Gueury, Michel
2005	Computers & Graphics	Technical section: Interpretive OpenGL for computer graphics	Chen, Bo ; Cheng, Harry H.
2006	Computer Graphics Forum	A PBL Experience in the Teaching of Computer Graphics	E. Martí, D. Gil, C. Julià
2006	SIGGRAPH Conference Proceedings	Teaching introductory computer graphics using java 3D, games and customized software: a Brazilian experience	Tori, Romero ; Bernardes, João Luiz ; Nakamura, Ricardo

2006	Technologies for E-Learning and Digital Entertainment	Mental vision: a computer graphics teaching platform	Peternier, Achille ; Thalmann, Daniel ;
2011	IEEE Computer Graphics and Applications	Teaching a Shader-Based Introduction to Computer Graphics	Angel, E ; Shreiner, D
2014	International journal of advanced manufacturing technology	A web service-based multi-disciplinary collaborative simulation platform for complicated product development	Ma, Songhua ; Tian, Ling
2016	American journal of pharmaceutical education	A Simulated Learning Environment for Teaching Medicine Dispensing Skills	Jenny McDowell ; Kim Styles ; Keith Sewell ; Peta Trinder ;
2017	IEEE Computer Graphics and Applications	The Case for Teaching Computer Graphics with WebGL: A 25-Year Perspective	Angel, Ed
2017	IEEE Transactions on Power Electronics	Questionnaire-Based Discussion of Finite Element Multiphysics Simulation Software in Power Electronics	X. Song ; V. Pickert ; B. Ji ; R. T. Naayagi ; C. Wang ; Y. Yerasimou
2018	COMPUTER GRAPHICS FORUM	Distinctive Approaches to Computer Graphics Education	Santos, Beatriz Sousa ; Dischler, Jean Michel ; Adzhiev, Valery ; Anderson, Eike Falk ; Ferko, Andrej ;
2018	MATEC web of conferences	Possibilities of applying structural optimization in building structures computer-aided design systems	Vasilkin, Andrey

2019	REM - International Engineering Journal	Numerical fundamentals and interactive computer graphics system for the nonlinear analysis of planar frames	Santana, Murillo VinÁ-cius Bento ; Silveira, Ricardo Azoubel da Mota
2019	International archives of the photogrammetry, remote sensing and spatial information sciences.	The integration of a scan-to-hbim process in bim application: the development of an add-in to guide users in autodesk revit	F. Banfi
2020	IEEE Access	Mesh Simplification Algorithm Based on the Quadratic Error Metric and Triangle Collapse	G. Zhou ; S. Yuan ; S. Luo
2020	IEEE Access	An Efficient Automatic Mesh Generator With Parabolic Arcs in Julia for Computation of TE and TM Modes for Waveguides	G. P. Kannan ; K. V. Nagaraja
2020	IEEE Computer Graphics and Applications	Looking Into a Future Which Hopefully Will Not Become Reality: How Computer Graphics Can Impact Our Behavior- A Study of the Potential of VR	Seurati, Giulia Wally ; Ferrise, Francesco ; Stork, Andre ; Basole, Rahul C.
2021	Computer Applications in Engineering Education	An interactive user interface for a structural analysis software using computer graphics techniques in MATLAB	Lopes, Pedro C. ; Rangel, Rafael L. ; Martha, Luiz F.

2021	IEEE Transactions on Visualization and Computer Graphics	A Suggestive Interface for Untangling Mathematical Knots	Liu, H. ; Zhang, H.
2021	Journal of Contemporary Issues in Business & Government	Designing the methodical system of the teaching process of computer graphics for the specialty of engineer-builder	Islomovna, Mamurova Feruza ; Ruzieva Dilnozalsomjonovna ; Khodjaeva Nodira Sharifovna;
2021	Education sciences	Computation and Learning Partnerships: Lessons from Wood Architecture, Engineering, and Construction Integration	Riggio, Mariapaola ; Cheng, Nancy Yen-wen


```

</button>
<button type="button" class="butoes">
  
</button>
<nav>
<div id="rightSide">
  <div id="divRightSide1">
    <input type="checkbox" id="grid"
    name="grid" checked="checked"
    onclick="checkGrid()">
    <label for="grid">Grid</label>
    <input type="checkbox" id="snap"
    name="snap" checked="checked"
    onclick="checkSnap()">
    <label for="snap">snap</label><br>
  </div> <div>
  <label id="PmouseX">x: </label>
  <label id="PmouseY">y: </label>
</div>
<!-- Material -->
<button type="button" class="butoes"
id="materialButton">
  
</button>
<div class="popup-wrapper">
  <div class="popup">
    <div class="popup-close"></div>
    <div class="popup-content">
      <h2>Material type</h2>
      <button type="button"
      class="butoes">
        
      </button>
      <button type="button"
      class="butoes">
        
      </button>
    </div>
  </div>
</div>

```

```

<button type="button"
class="butoes">
  
  </button><br><br>
  <h3>other type of
material</h3><br>
  <button type="button"
class="teste">
    <input type="text"
id="outrosButton" style="text-align:center"
class="outros-input" placeholder="Enter a
value"><br><br>
    <input type="submit"
onclick="outrosButton()" value="SUBMIT"
class="outros-submit" onsubmit="return
validarMaterial()">
  </button> </div> </div> </div>
  <main id="_main" >
    <script src="sketch.js"></script>
    <script src="myModel.js"
type="text/javascript"></script>
    <script src="myView.js"
type="text/javascript"></script>
    <script src="myController.js"
type="text/javascript"></script>
  </script src="myController.js"
type="text/javascript"></script>
  <script src="mouseEvents.js"
type="text/javascript"></script>
  <script src="Line.js"
type="text/javascript"></script>
  <script src="Polyline.js"
type="text/javascript"></script>
  <script src="Point.js"
type="text/javascript"></script>
  <script src="nodeClass.js"
type="text/javascript"></script>
  <script src="elementClass.js"
type="text/javascript"></script>
  <script src="materialClass.js"
type="text/javascript"></script>
  <script src="StructureClass.js"
type="text/javascript"></script>
  <script src="crossSectionClass.js"
type="text/javascript"></script>
  <script src="supportClass.js"
type="text/javascript"></script>
  <script src="loadClass.js"
type="text/javascript"></script>
  <script src="drawClass.js"
type="text/javascript"></script>
</main>
<div id="leftSide">
  <!-- Check Box-->
  <button type="button" class="box1">
    
    </button><br><br>
    <button type="button" class="box2">
      
      </button><br><br>
      <button type="button" class="box3">
        
        </button>
      </div>
    <footer>Footer</footer>
  </body>
</html>

```

Código fonte – StructureClass

```

class structure {
  constructor() {
    this.type = 'TP2'
    this.nnos = 0 //nodes number of the
    structre
  }
}

```

```

this.nmats = 0 //materials number of
the structre

this.nsecs = 0 //cross sections number
of the structre

this.nelems = 0 //elements number of
the structre

this.naps = 0 //support number of the
structre

this.nnoscar = 0 //nodes loaded
number of the structre

this.nelemscar = 0 //elements number
of the structre

this.neq = 0 //number of equations to
solve problem

//cordinates of each node to compose
structre

//the quantity of line of this matrix
have to be the same number of the nnos

this.matrixOfNodes = math.matrix()
//store nodes objects

this.coordnos = math.matrix()
//matrix of the material properties that
compose the structre

//the quantity of line of this matrix
have to be the same number of the nmats

this.matrixOfSup = math.matrix()
//store supports objects

this.restrsap = math.matrix()
//this matrix defines the distributed
load of each element

//the quantity of line of this matrix
have to be the same number of the nnoscar

this.matrixOfNosCar = math.matrix()
//store nodes loaded objects

this.cargasnos = math.matrix()
//this matrix defines what node of
each element have a pontual load

//the quantity of line of this matrix
have to be the same number of the
nelemscar

this.cargaselems = math.matrix()
//degree of freedom matrix of the
node

this.glnos = []
//Global stiffness matrix of the
structure

this.Kestr = math.matrix()
//External loads of the structure

this.Festr = math.matrix()

```

```

//Nodal displacement
this.Uestr = math.matrix(0) }
##### 1- LEITURA DE
DADOS#####
//Creating a new node and adding in the
matrix of coordinates to compose structure
createNode(posX, posY) {
//creating a new line to set the new
node position of the structure
this.coordnos =
math.subset(this.coordnos,
math.index(this.nnos, [0, 1]), [posX,
posY])
//add number of nodes
this.nnos++
//create a new node object and adding
in the matrix of nodes
this.matrixOfNodes[this.nnos - 1] =
new node(posX, posY, this.nnos)
} //Creating a new material and adding
in the matrix of materials to compose
structure
createMaterial(type) {
//add number of material
this.nmats++
}
//create a new material object
this.matrixOfMat[this.nmats - 1] =
new material(this.nmats)
switch (type) {
case 'TP2':
//creating a new line to set the
new material position of the structure
this.propmats =
math.subset(this.propmats,
math.index(this.nmats - 1, [0, 1]), [0, 0])
break;
default:
break;
}
//This method is used to create new cross
section
//and set in the appropriated matrix
createCrossSection(type) {
//add number of cross Section
this.nsecs++
//create a new cross Section object
and it is stored in a matrix
this.matrixOfcrossSec[this.nsecs - 1]
= new crossSection(type, this.nsecs)
switch (type) {
case 'TP2':
//creating a new line to set the
new cross elements of the structure
this.propelems =
math.subset(this.propelems,
math.index(this.nsecs - 1, [0]), 0)
break;
default:
break; } //This method
is used to create new elements
//and set in the appropriated matrix
createElements(type) {
//add number of element
this.nelems++
//create a new element object and it is
stored in a matrix
this.matrixOfelems[this.nelems - 1] =
new element(type, this.nelems)
switch (type) {
case 'TP2':
//creating a new line to set the
new cross elements of the structure
//%__no1__no2__material_seção
this.propelems =
math.subset(this.propelems,

```



```

#"lembre-se", a matriz de restrições
restrap tem 3 colunas, onde se define para
cada nó sua restrição, ou seja, 0 ou 1

#então, a primeira coluna de restrasp
não entra na matriz glno, porque é a coluna
dos nós.
*/      var nglno =
this.matrixOfelems[0].nglno

this.glno = math.zeros(this.nnos,
nglno)
for (var i = 0; i < this.naps; i++) {
var no =
this.restrsap.subset(math.index(i, 0))
for (var j = 1; j <= this.naps; j++) {
var a =
this.restrsap.subset(math.index(i, j))
this.glno.subset(math.index(no -
1, j - 1), a)
} } for (var k = 0; k <
this.nnos; k++) {
for (var p = 0; p < nglno; p++) {
var b =
this.glno.subset(math.index(k, p))
if (b == 1) {
this.glno.subset(math.index(k,
p), 0)
} else {
this.neq++
this.glno.subset(math.index(k,
p), this.neq) } } }
##### 2 - CRIAÇÃO
DA MATRIZ DE GRAUS DE
LIBERDADE DA
ESTRUTURA#####
#####
##### 3 - CRIAÇÃO
DA MATRIZ DE RIGIDEZ DA
ESTRUTURA#####
#####
stiffnessMatrixOfStruct() {
//cria o tamanho do vetor de graus de
liberdade do elemento, por exemplo, se
nglno = 2 (graus de liberdade do elemento)
//e nnoselem = 2 (número de nós por
elemento), então o número de graus de
liberdade por elemento será 4.
var glno = this.glno
var nglno =
this.matrixOfelems[0].nglno
var nnoselem =
this.matrixOfelems[0].nnoselem
var nglel = nglno * nnoselem
} else {
this.neq++
this.matrixOfelems[0].npropmat
var npropgeo =
this.matrixOfelems[0].npropgeo
var type = this.matrixOfelems[0].type
//Assim, este vetor irá guardar, para o
exemplo anterior, 4 graus de liberdade,
sendo 2 graus por nó.
//Portanto, o vetor gle = (gdl x do nó
1, gdl y do nó 1, gdl x do nó 2, gdl y do nó
2) totalizando 4 posições para o exemplo
anterior var gle = math.zeros(1, nglel)
//Aqui é criada a matriz de rigidez
global neq por neq, ou seja, nesta matriz já
foram retiradas as linhas e colunas dos
apoios, quando for encontrar os
deslocamentos
var neq = this.neq
this.Kestr = math.zeros(neq, neq)
//este laço irá criar a matriz de rigidez
global for (var el = 0; el < this.nelems;
el++) {
//aqui se identifica o material do
elemento
var idmat =
this.propelems.subset(math.index(el,
nnoselem))

```

```

//aqui se identifica a seção do
elemento
var idsec =
this.propelems.subset(math.index(el,
nnoselem + 1))
//aqui pega toda a linha do material
daquele elemento
var pmat = math.zeros(1, npropmat)
pmat =
this.propmats.subset(math.index(idmat - 1,
math.range(0, npropmat + 1)))
//aqui pega toda a linha da
propriedade da seção transversal do
elemento
var psec = math.zeros(1, npropgeo)
psec =
this.propgeo.subset(math.index(idsec - 1,
math.range(0, npropgeo + 1)))
//Aqui está chamado de "no", mas
na verdade é a barra que vai do nó da
posição 1 até a posição 4 do vetor "no"
var no =
this.propelems.subset(math.index(el,
math.range(0, nnoselem)))
if (type == 'TP2' || type == 'PP2' ||
type == 'GP2') {
//Cálculo de L, cs e sn

```

```

//variáveis para calcular a
distância entre nós
var no1 =
no.subset(math.index(0, 0))
var no2 =
no.subset(math.index(0, 1))
//coletando os valores das
coordenadas x e y de cada nó
var x1 =
this.coordnos.subset(math.index(no1 - 1,
0))
var x2 =
this.coordnos.subset(math.index(no2 - 1,
0))
var y1 =
this.coordnos.subset(math.index(no1 - 1,
1))
var y2 =
this.coordnos.subset(math.index(no2 - 1,
1))
//calculando as componentes x e
y da distância entre os nós
var dx = x2 - x1
var dy = y2 - y1
//calculando a distância por
pitágoras

```

```

var L = math.sqrt(dx * dx + dy *
dy)
//calculando o cosseno e seno da
estrutura
var cs = dx / L
var sn = dy / L
if (L <= 0.00000001) {
alert('Error when trying to
evaluate the distance of element ${el + 1},
value near to zero')
break } else {
alert("This element type is not
implemented, for this is impossible to
create matrix of the structre, erro in the
rigidMatrixOfStructre()")
break;
}
//Creating the local stiffness and
rotational matrix
if (type == 'TP2') {
//setting material and cross
section of the element "el" to create the
stiffness matrix
var mat =
pmat.subset(math.index(0, 0))

```

```

var sec =
psec.subset(math.index(0, 1))

this.matrixOfelems[e].E = mat
this.matrixOfelems[e].A = sec
this.matrixOfelems[e].L = L
//Creating the local stiffness
matrix

this.matrixOfelems[e].stiffnessMatrix()

var Kel =
math.subset(this.matrixOfelems[e].Kel,
math.index(math.range(0, nglel),
math.range(0, nglel)))
//Creating the rotational matrix
this.matrixOfelems[e].cs = cs
this.matrixOfelems[e].sn = sn

this.matrixOfelems[e].rotationalMatrix()
var Rel =
math.subset(this.matrixOfelems[e].Rel,
math.index(math.range(0, nglel),
math.range(0, nglel))) //
.subset(math.index(math.range(0, nglel),
math.range(0, nglel))) } else {
alert(`Matrix of this element was
not implemented, please verify the element
${e} + 1`)
}

break } //transferring from
local sistem to the lobal sistem
if (type != 'ISO') { Kel =
math.multiply(math.multiply(math.transpo
se(Rel, Kel), Rel) }
//Degree of freedom (DOF) for
compose element in the global stiffness
matrix
gle =
this.matrixOfelems[e].DOFelement(glno,
no, nglno)
//setting matrix element in the
global stiffness matrix for compose
structure
var p
var k
var Kini
var ke
for (var i = 0; i < nglel; i++) {
if (gle[0][i] > 0) {
for (var j = 0; j < nglel; j++) {
if (gle[0][j] > 0) {
p = gle[0][i] - 1
k = gle[0][j] - 1
}
}
}
}

Kini =
this.Kestr.subset(math.index(p, k))
ke =
this.Kestr =
math.subset(this.Kestr, math.index(p, k),
Kini + ke)
} } } }
##### 3 - CRIAÇÃO
DA MATRIZ DE RIGIDEZ DA
ESTRUTURA#####
#####
##### 4 - FORÇAS
EXTERNAS DA
ESTRUTURA#####
#####
externalLoads() {
var nglno =
this.matrixOfelems[0].nglno
this.Festr = math.zeros(this.neg, 1)
//Extenal forces in the nodes
var node
var gl
var Fini
var F
for (var i = 0; i < this.mnoscar; i++) {
}
}

```



```

console.log('Coordenadas dos nós')
console.log('%__X__Y')
console.log(this.coordnos)
console.log('Propriedades dos
materiais')

console.log('%__E__ALPHA
')

console.log(this.propmats)

console.log('Propriedades_das_seções_tran
sversais')

console.log('%__Area')
console.log(this.propgeo)

console.log('Propriedades_dos_elementos')

console.log('__no1__no2_material_seção'
)

console.log(this.propelems)

console.log('Apoios_(r=1_impedido_r=0_li
vre)')

console.log('__no__rtx_rty')
console.log(this.restsap)

console.log('Nos_carregados')
console.log('__no__fx__fy')
console.log(this.cargasnos)
##### PRINTA
LEITURA DE
DADOS#####
#####
##### PRINTA
CRIAÇÃO DA MATRIZ DE GRAUS DE
LIBERDADE DA
ESTRUTURA#####
console.log('Número de graus de
liberdade do nó')

console.log(this.matrixOfelems[0].nglno)
console.log('Matrix de graus de
liberdade do nó')
console.log(this.glno)
console.log('Número de equações')
console.log(this.neq)
##### PRINTA
CRIAÇÃO DA MATRIZ DE GRAUS DE
LIBERDADE DA
ESTRUTURA#####
##### PRINTA
CRIAÇÃO DA MATRIZ DE RIGIDEZ
DA
ESTRUTURA#####
console.log('Matriz de Rigidez da
estrutura')
console.log(this.Kestr)
##### PRINTA
CRIAÇÃO DA MATRIZ DE RIGIDEZ
DA
ESTRUTURA#####
##### PRINTA
FORÇAS EXTERNAS DA
ESTRUTURA#####
console.log('Forças externas')
console.log(this.Festr)
##### PRINTA
FORÇAS EXTERNAS DA
ESTRUTURA#####
##### PRINTA
DESLOCAMENTO NODAL DA
ESTRUTURA#####
console.log('Deslocamento nodal da
estrutura')

```

```

console.log(this.Uestr)
##### PRINTA
DESLOCAMENTO NODAL DA
ESTRUTURA#####
#####
}
##### 6 - PRINTA
OS REUSLTADOS NO
CONSOLE#####
###
##### 7 - OUTROS
COMANDOS - COMO RESET
MESH#####
resetMest() {
  this.nnos = 0 //nodes number of the
  structre
  this.nmats = 0 //materials number of
  the structre
  this.nsecs = 0 //cross sections number
  of the structre
  this.nelems = 0 //elements number of
  the structre
  this.naps = 0 //support number of the
  structre
  this.nnoscar = 0 //nodes loaded
  number of the structre

  this.nelemscar = 0 //elements number
  of the structre
  this.neg = 0 //number of equations to
  solve problem
  //coordinates of each node to compose
  structre
  //the quantity of line of this matrix
  have to be the same number of the nnos
  this.matrixOfNodes = math.matrix()
  //store nodes objects
  this.coordnos = math.matrix()
  //matrix of the material properties that
  compose the structre
  //the quantity of line of this matrix
  have to be the same number of the nmats
  this.matrixOfMat = math.matrix()
  //store material objects
  this.propmats = math.matrix()
  //matrix of the cross section of each
  element that compose the structre
  //the quantity of line of this matrix
  have to be the same number of the nsecs
  this.matrixOfcrossSec = math.matrix()
  //store cross section objects
  this.propgeo = math.matrix()

  //matrix that defines what nodes
  compose one element
  //define the what the line of propmats
  and propgeo is the material and cross
  section of that element
  //the quantity of line of this matrix
  have to be the same number of the nelems
  this.matrixOfelems = math.matrix()
  //store elements objects
  this.propelems = math.matrix()
  //this matrix defines the supports that
  compose the structure
  //the quantity of line of this matrix
  have to be the same number of the naps
  this.matrixOfsup = math.matrix()
  //store supports objects
  this.restrsap = math.matrix()
  //this matrix defines the distributed
  load of each element
  //the quantity of line of this matrix
  have to be the same number of the nnoscar
  this.matrixOfNosCar = math.matrix()
  //store nodes loaded objects
  this.cargasnos = math.matrix()

```

```

//this matrix defines what node of
each element have a pontual load
//the quantity of line of this matrix
have to be the same number of the
nelemscar
    this.cargaselems = math.matrix() }
##### 7 - OUTROS
COMANDOS - COMO RESET
MESH#####
}
Código fonte – Mycontroler
class myControler {
    constructor(view) {
        //mouse position
        this.mouseInCanvas = true
        this.precision = 1
        this.dx = round(0.0, this.precision)
        this.dy = round(0.0, this.precision)
        this.cursor2D = cursor(CROSS)
        this.view = view
        this.isSnap = true
        this.gridX = 1
        this.gridY = 1
        this.isGrid = true
        //Line Button
        this.statusLineButton = false
        this.typeClickLine = "
        //curve type
        this.curveType = "
        //point button
        this.statusPointButton = false
        this.PmouseX =
        document.getElementById("PmouseX")
        this.PmouseY =
        document.getElementById("PmouseY")
        //structure object
        this.elementIdentiicator = []
        this.struct = new structure()
        //displacement results
        this.drawDisplacementResult = false //flag
        this.displacementVectorControl = []
        //vector de resultados de deslocamento }
        show() {
            background(204, 227, 232);
            //background(255, 0, 0);
            if (this.isGrid) {
                this.makeDisplaygrid(view) }
            if (_objectOfDrawing.isDrawing) {
                _objectOfDrawing.drawRectangleOfSelection(
                )
                _objectOfDrawing.drawForce()
                _objectOfDrawing.drawSupport()
                if (this.drawDisplacementResult) {
                    _objectOfDrawing.displacementResults()
                }
                _model.showModel()
                view.camera() }
                //position in the new scale with origin in the
                same position of the camera world
                mousePosition() {
                    this.dx = round((mouseX / width) *
                    (view._right - view._left) + view._left,
                    this.precision)
                    this.dy = round((mouseY / height) *
                    (view._bottom - view._top) + view._top),
                    this.precision)
                    if (this.mouseInsideCanvas()) {
                        //snap
                        if (this.isSnap) {
                            let snap = []
                            snap = this.snapTo()
                            this.dx = snap[0].x
                            this.dy = snap[0].y
                        }
                    }
                }
            }
        }
    }
}

```

```

//drawing in the label
this.PmouseX.textContent = `X:
${this.dx}`
this.PmouseY.textContent = `Y:
${this.dy}` } }
mouseCursor(type) {
  this.cursor2D = cursor(type)
  //colocar estes cursores na situação
  escolhida.
  /* this.cursor2D = cursor(
this.cursor2D = cursor(ARROW)
this.cursor2D = cursor(CROSS)
this.cursor2D = cursor(HAND)
this.cursor2D = cursor(MOVE)
this.cursor2D = cursor(TEXT)
this.cursor2D = cursor(WAIT) */ }
mouseInsideCanvas() {
  if (this.dx >= this.view._right
  || this.dx <= this.view._left ||
  this.dy >= this.view._top ||
  this.dy <= this.view._bottom) {
    this.mouseInCanvas = false
  } else {
    this.mouseInCanvas = true
  }
  return this.mouseInCanvas
}
origin(oX, oY) {
//create origin
push()
let oX = 0
let oY = 0
let x = 0
let y
//points to make a grid
beginShape(POINTS)
stroke(0)
strokeWeight(1.5)
while (x >= view._left) {
  y = 0
  while (y >= view._bottom) {
    vertex(x, -y)
    y -= this.gridY
  }
  x -= this.gridX
}
x = 0
while (x <= view._right) {
  y = 0
  while (y >= view._bottom) {
    vertex(x, -y)
    y -= this.gridY
  }
  x += this.gridX
}
x = 0
while (x <= view._right) {
  makeDisplaygrid(view) {

```

```

y = 0
while (y <= view._top) {
  vertex(x, -y)
  y += this.gridY
  x += this.gridX
}
x = 0
while (x >= view._left) {
  y = 0
  while (y <= view._top) {
    vertex(x, -y)
    y += this.gridY
  }
  x -= this.gridX
}
endShape()
this.origin(oX, oY)
pop()
setGrid(gridx, gridy) {
  this.gridX = gridx
  this.gridY = gridy
}
getGrid() {
  let grid = []
  grid.push(createVector(this.gridX,
    this.gridY))
  return grid
}
setSnap(booleanaSnap) {
  this.isSnap = booleanaSnap
}
snapTo() {
  let ip; // integer part
  let fp; // "fraction" part
  let x = this.dx;
  let y = this.dy;
  let snap = []
  fp = x / this.gridX;
  ip = parseInt(fp);
  fp = fp - ip;
  if (fp > 0.5)
    x = (ip + 1.0) * this.gridX;
  else if (fp < -0.5)
    x = (ip - 1.0) * this.gridX;
  else
    x = ip * this.gridX;
  fp = y / this.gridY;
  ip = parseInt(fp);
  fp = fp - ip;
  if (fp > 0.5)
    y = (ip + 1.0) * this.gridY;
  else if (fp < -0.5)
    y = (ip - 1.0) * this.gridY;
  else
    y = ip * this.gridY;
  snap.push(createVector(x, y))
  return snap
}
//methods to create curves
createCurvesToDraw() {
  switch (this.curveType) {
    case 'LINE':
      _model.numberOfcurves++
      _model.m_curves.push(new
        _line(_model.numberOfcurves))
      _model.m_curves[_model.m_curves.length -
        1].currentCurve = true
      noFill()
      break;
    case 'POLYLINE':
      _model.numberOfcurves++
      _model.m_curves.push(new
        _polyline(_model.numberOfcurves))
      _model.numberOfcurves++
      _model.m_curves[_model.m_curves.length -
        1].currentCurve = true
      noFill()
      break;
  }
}

```



```

if (_model.m_curves.length >= 1) {
  for (var i = 0; i <
    _model.m_curves.length; i++) {
    let pts = []
    pts = _model.m_curves[i].getPoints()
    for (var j = 0; j < pts.length; j++) {
      _model.m_verts.push(createVector(pts[j].x, -
        pts[j].y))
      this.createVectorOfobjectPoints(j, -
        pts) } } }
//cria objetos de pontos
create VectorOfobjectPoints(j, pts) {
  _model.numberOfcurvers++
  _model.m_points.push(new
    point(_model.numberOfcurvers))
  _model.m_points[j].m_x1 = pts[j].x
  _model.m_points[j].m_y1 = pts[j].y
} //verifica se no vetor de pontos m_verts
possui pontos iguais. se houver elimina o igual.
verifyIfPointsEqual() {
  //vetor auxiliar
  var aux = []
  //verifica se algum ponto é igual ao outro.
  if (_model.m_curves.length >= 1) {
    for (var i = 0; i <
      _model.m_verts.length; i++) {
      var isEqual = false
      if (i < _model.m_verts.length - 1) {
        for (var j = i + 1; j <
          _model.m_verts.length; j++) {
          if (i != j) {
            if (_model.m_verts[i].x ==
              _model.m_verts[j].x &&
              _model.m_verts[i].y ==
              _model.m_verts[j].y) {
              isEqual = true
              break } }
            //se o ponto for diferente dos
            anteriores então o ponto é excluído do vetor
            auxiliar
            aux.push(createVector(_model.m_verts[i].x,
              _model.m_verts[i].y))
            if (isEqual == false) {
              //se o vetor auxiliar
              tiver tamanho diferente dos vetor verts então
              algum ponto foi excluído. Neste sentido o vetor
              m_verts é criado novamente sem o ponto que
              era igual.
              if (aux.length !=
                _model.m_verts.length) {
                _model.m_verts = []
              }
            }
          }
        }
      }
    }
    pts.push(createVector(aux[j].x, -
      aux[j].y))
    this.createVectorOfobjectPoints(j, pts)
  } } }
addTypeOfElement() {
  return 'TP2' }
createMaterial(MaterialType) {
  var outro =
    document.querySelector("#outrosButton");
  var valor = outro.value;
  console.log(valor);
  switch (MaterialType) {
    case 'CONCRETO':
      this.struct.createMaterial(this.struct.type)
    this.struct.matrixOfMat[this.struct.nmats -
      1].materialTruss(this.struct, this.struct.nmats -
      1, 23000, 0.00010)
    break;
    case 'AÇO':

```

```

this.struct.createMaterial(this.struct.type)

this.struct.matrixOfMat[this.struct.nmats -
1].materialTruss(this.struct, this.struct.nmats -
1, 200000, 0.000011)
    break;
    case 'ALUMÍNIO':
this.struct.createMaterial(this.struct.type)
this.struct.matrixOfMat[this.struct.nmats -
1].materialTruss(this.struct, this.struct.nmats -
1, 60000, 0.000022)
    break;
    case 'OUTROS':
this.struct.createMaterial(this.struct.type)
###complete aqui
        ###complete aqui
this.struct.matrixOfMat[this.struct.nmats -
1].materialTruss(this.struct, this.struct.nmats -
1, valor, 0.000022)
    break;
    default:
        console.log('Material inexistente');
    }
this.struct.matrixOfMat[this.struct.nmats -
1].name = MaterialType
    console.log('Material adicionado')
} //configurar o material em um elemento

setMaterial(material) {
    if (this.statusMaterialButton) {
        for (var j = 0; j <
_model.m_curves.length; j++) {
            if (_model.m_curves[j].selected) {
                material =
_model.m_curves[j].material =
                secaoTraversal() {
                    this.struct.createCrossSection(this.struct.type)
                    this.struct.matrixOfcrossSec[this.struct.nsecs -
1].crossSectionTruss(this.struct, 100)
                    console.log('Seção transversal
adicionada') }
                //configura o material na curva da geometria
                setCrossSection(crossSection) {
                    if (this.statusSectionButton) {
                        for (var j = 0; j <
_model.m_curves.length; j++) {
                            if (_model.m_curves[j].selected) {
                                = crossSection
                                CreateSupport() {
                                    //criando flag para desenho do suportes
                                    for (var j = 0; j < _model.m_points.length;
j++) {
                                        if (_model.m_points[j].selected) {
                                            _model.m_points[j].drawSupport = true
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
                //sconfigura o suporte no ponto da geometria
                setSupport(crossSection) {
                    if (this.statusSupportButton) {
                        for (var j = 0; j <
_model.m_points.length; j++) {
                            if (_model.m_points[j].selected) {
                                crossSection
                                }
                            }
                        }
                    }
                }
                createForce() {
                    //criar o método de força
                    //criando flag para desenho do suportes
                    for (var j = 0; j < _model.m_points.length;
j++) {
                        if (_model.m_points[j].selected) {
                            _model.m_points[j].drawForce = true
                        }
                    }
                }
            }
        }
    }
}

```



```

//este laço varre os elementos para criar
suportes,
for (var i = 0; i <
_model.m_points.length; i++) {
if (_model.mesh) {
//cria o support do elemento
//criando suportes da malha
//cria o support no processador
para criação da estrutura
if
(_model.m_points[i].drawSupport) {
let rtx
let rty
let rmz
switch
(_model.m_points[i].supportType) {
case 'PRIMEIROGENERO':
rtx = 1
rty = 0
rmz = 0
break
case 'SEGUNDOGENERO':
rtx = 1
rty = 1
rmz = 0
break
case 'TERCEIROGENERO':
break
rtx = 1
rty = 1
rmz = 1
break
default:
window.alert('SUPPORT
INEXISTENTE');
this.struct.createSupport(this.struct.type)
this.struct.matrixOfsup[this.struct.naps -
1].setSupportNode(this.struct, this.struct.naps -
1, this.struct.matrixOfNodes[i], rtx, rty)
}
//cria a força no processador para
criação da estrutura
if (_model.m_points[i].drawForce)
{
this.struct.createLoads('nodal')
this.struct.matrixOfNosCar[this.struct.nnoscar
- 1].nodalLoad(this.struct,
this.struct.matrixOfNodes[i],
this.struct.nnoscar - 1, 0, -100000)
}
} else {
_model.geometry = true
_model.mesh = false
this.struct
this.struct.resetMest()
}
redraw()
runMesh() {
//CONFERÊNCIA DE ERROS
if (this.struct.nnos < 2) {
window.alert('Necessário ao menos
dois nós')
return
}
if (_model.m_curves.length < 1) {
window.alert('Necessário ao menos a
criação de uma geometria')
return
}
if (this.struct.nsecs < 1) {
window.alert('Necessário criar ao
menos uma seção transversal')
return
}
if (this.struct.nmats < 1) {
window.alert('Necessário criar ao
menos um material')
return
}
if (this.struct.nelems < 1) {

```



```
this.displacementVectorControl.push(createVecto  
r(dx, dy)) } }
```

Código fonte – MyModel

```
class myModel {
  constructor() {
    //vector of points
    this.m_verts = [] //armazenas as
    coordenadas de pontos para facilitar na
    criação dos elementos
    this.m_points = [] //armazenas os
    objetos de pontos
    this.m_vcolors = []
    this.m_curves = [] //armazena as
    curvas em geral
    this.numberOfcurvers = 0
    //flag
    this.geometry = true
    this.mesh = false }
  showModel() {
    if (this.m_curves.length >= 1) {
      for (var i = 0; i <
        this.m_curves.length; i++) {
        push(
          vertex(this.m_verts[i].x,
            this.m_verts[j].y);
          endShape(); } }
        pointSelected(j) {
          if (this.geometry) {
            if (this.m_points[j].selected) {
              stroke('red')
            } else {
              stroke('black') } }
            if (this.mesh) {
              if (this.m_points[j].selected) {
                stroke('red')
              } else {
                stroke('black') } }
              deleteShape() {
                for (var i = 0; i <
                  this.m_curves.length; i++) {
                  print()
                  if (this.m_curves[i].isSelected) {
                    this.m_curves[i].pop()
                  }
                }
                deletePoint(positionPoint) {
                  var vector = []
                  var j = 0
                }
              }
            }
          }
        }
      }
    }
  }
}
```

```

for (var i = 0; i <
this.m_curves.length; i++) {
    if(positionPoint != i){
        vector[j] = this.m_verts[i]
        j++
    }
    this.m_verts = []
    for (var i = 0; i <
this.m_curves.length; i++) {
        this.m_verts[i] = vector[i]
    }
    getBoundingBox(_left, _right, _bottom,
_top) {
        let boundingBoxVector = []
        if (this.m_verts.length < 1 ||
this.m_curves.length < 1) {
            let _xmin = -1
            let _xmax = 9
            let _ymin = 1
            let _ymax = -9

            boundingBoxVector.push(createVector(_xmin
, _ymin))

            boundingBoxVector.push(createVector(_xmin
, _ymax))

            return boundingBoxVector
        }
Código fonte – MyView
class myView {
    constructor(model) {
        //camera attributes
        this.x = 0
        this.y = 0
        this.z = (height / 2) / (PI / 6)
        this.centerX = 0
        this.centerY = 0
        this.centerZ = 0
        this.upX = 0
        this.upY = 1
        this.upZ = 0
        //projection attributes
        this.projection
        //ortho
        this._left = -1
        this._right = 9
        this._bottom = -1
        this._top = 9
        this._near = 0.1
        this._far = 1000
        //perspective
        this.fovy = PI / 3.0
    }
    return boundingBoxVector;
}
let _xmin = this.m_verts[0].x
let _xmax = _xmin
let _ymin = this.m_verts[0].y
let _ymax = _ymin
for (var i = 1; i < this.m_verts.length;
i++) {
    if (this.m_verts[i].x < _xmin)
        _xmin = this.m_verts[i].x;
    if (this.m_verts[i].x > _xmax)
        _xmax = this.m_verts[i].x;
    if (this.m_verts[i].y > _ymin)
        _ymin = this.m_verts[i].y;
    if (this.m_verts[i].y < _ymax)
        _ymax = this.m_verts[i].y;
}
boundingBoxVector.push(createVector(_xmin
, _ymin))

boundingBoxVector.push(createVector(_xmin
x, _ymax))

return boundingBoxVector
}

```

```

this.aspect = width / height
//movement attributes
this._space = 5
this.cam = createCamera(this.x, this.y,
this.z,
    this.centerX, this.centerY,
    this.upX, this.upY, this.upZ)
this.cam.ortho(this._left, this._right,
this._bottom,
    this._top, this._near, this._far);
//loop
this.angle = 0
//set model    this.model = model
}
setOrthoValues(bounding) {
    this._left = bounding[0].x
    this._right = bounding[1].x
    this._bottom = -bounding[0].y
    this._top = -bounding[1].y }
camera(projection) {
    //setting projection
    if (projection == perspective) {
        this.cam.perspective(this.fovy,
this.aspect, this._near, this._far)
    } else {
        this.cam.setPosition(this.x, this.y,
this.z)
        this.cam.lookAt(this.centerX,
this.centerY, this.centerZ)
        this.cam.ortho(this._left, this._right,
this._bottom,
            this._top, this._near, this._far);
    }
    panWorldWindow(_panFacX,
_panFacY) {
        let panX = abs(this._right - this._left)
        * _panFacX
        let panY = abs(this._top -
this._bottom) * _panFacY
        //shift window
        this._left += panX
        this._right += panX
        this._top += panY
        this._bottom += panY
        redraw() }
        moving() {
            this.cam.move(65) || keyIsDown(37))
            { // tecla A or left arrow
                //this.cam.move(this._space, 0, 0);
                this.panWorldWindow(0.1, 0)
            }
            if (keyIsDown(68) || keyIsDown(39))
            // tecla D or right arrow {
                //this.cam.move(-this._space, 0, 0);
                this.panWorldWindow(-0.1, 0)
            }
            if (keyIsDown(87) || keyIsDown(38))
            { // tecla W or up arrow
                //this.cam.move(0, this._space, 0);
                this.panWorldWindow(0, -0.1)
            }
            if (keyIsDown(83) || keyIsDown(40))
            // tecla S or down arrow {
                //this.cam.move(0, -this._space, 0);
                this.panWorldWindow(0, 0.1) }
        scaleWorldWindow(_scaleFac) {
            var vpr // viewport distortion
            ratio
            var cx, cy // window center
            var sizex, sizey // window sizes

```

```

var Lx, Ly
//Compute canvas viewport ratio.
vpr = height / width
// Get current window center.
cx = (this._left + this._right) / 2.0
cy = (this._bottom + this._top) / 2.0
// Set new window sizes based on
scaling factor.

this._left = (cx - Lx / 2.0)
this._right = (cx + Lx / 2.0)
this._bottom = (cy - Ly / 2.0)
this._top = (cy + Ly / 2.0)
// Establish the clipping volume by
setting up an
// orthographic projection
this.camera()
redraw() }
fitWorldToViewport(_scaleFac) {
let bounding =
this.model.getBoundingBox(this._left,
this._right, this._bottom, this._top)
this.setOrthoValues(bounding)

sizey = (this._left - this._right) *
_scaleFac
sizey = (this._top - this._bottom) *
_scaleFac
// Adjust window to keep the same
aspect ratio of the viewport.

if ((sizey / sizez) < vpr) {
Ly = sizey
Lx = Ly / vpr }
else if ((sizey / sizez) > vpr) {
Lx = sizez
Ly = Lx * vpr }

buleanLoop = true
buttonLoop.html('noLoop')
angle = 0
this.x = (height / 2) / (PI / 6) *
sin(angle)
this.z = (height / 2) / (PI / 6) *
cos(angle)
loop() } else {
buleanLoop = false
buttonLoop.html('Loop')
angle = 0
this.x = (height / 2) / (PI / 6) *
sin(angle)
this.z = (height / 2) / (PI / 6) *
cos(angle)
}

this.scaleWorldWindow(_scaleFac)
scrollDrawing(_scroll) {
_scale = _scale + _scroll
if (_scale > 0) {
this._left = -width / _scale
this._right = width / _scale
this._bottom = -height / _scale
this._top = height / _scale }
else {
_scale = abs(_scroll)
redraw() }
useLoop() }
if (buleanLoop == false) {

```

```

noLoop()
  redraw() } }
//make the model rotate around y axis
animateModel() {
  if (buleanLoop) {
    this.x = (height / 2) / (PI / 6) *
sin(this.angle)
    this.z = (height / 2) / (PI / 6) *
cos(this.angle)
  }
  this.angle += 0.01
  this.moving() } }

```