



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**INSTITUTO UNIVERSIDADE VIRTUAL**  
**CURSO DE BACHARELADO EM SISTEMAS E MÍDIAS DIGITAIS**

**MATHEUS RODRIGUES DA SILVA**

**GNUPLOT ON WEB (GOW): UMA APLICAÇÃO PARA USO DAS**  
**FUNCIONALIDADES DO GNUPLOT EM FÓRUNS DE AMBIENTES VIRTUAIS DE**  
**APRENDIZAGEM**

**FORTALEZA**

**2021**

MATHEUS RODRIGUES DA SILVA

GNUPLOT ON WEB (GOW): UMA APLICAÇÃO PARA USO DAS FUNCIONALIDADES  
DO GNUPLOT EM FÓRUMS DE AMBIENTES VIRTUAIS DE APRENDIZAGEM

Trabalho de Conclusão de Curso apresentado  
ao Curso de Graduação em Sistemas e Mídias  
Digitais da Universidade Federal do Ceará,  
como requisito parcial à obtenção do grau de  
Bacharel em Sistemas e Mídias Digitais.

Orientador: Prof. Me. Wellington Wagner  
Ferreira Sarmiento.

FORTALEZA

2021

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

---

S581g Silva, Matheus Rodrigues da.  
Gnuplot on web (GOW) : uma aplicação para uso das funcionalidades do gnuplot em fóruns de ambientes virtuais de aprendizagem / Matheus Rodrigues da Silva. – 2021.  
31 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Instituto UFC Virtual, Curso de Sistemas e Mídias Digitais, Fortaleza, 2021.  
Orientação: Prof. Me. Wellington Wagner Ferreira Sarmento.

1. Plotagem de gráficos. 2. Web editor. 3. Ambiente Virtual de Aprendizagem. I. Título.

CDD 302.23

---

MATHEUS RODRIGUES DA SILVA

WEBPLOT GNUPLOT ON WEB (GOW): UMA APLICAÇÃO PARA USO DAS  
FUNCIONALIDADES DO GNUPLOT EM FÓRUMS DE AMBIENTES VIRTUAIS DE  
APRENDIZAGEM

Trabalho de Conclusão de Curso apresentado  
ao Curso de Graduação em Sistemas e Mídias  
Digitais da Universidade Federal do Ceará,  
como requisito parcial à obtenção do grau de  
Bacharel em Sistemas e Mídias Digitais.

Aprovada em: \_\_\_/\_\_\_/\_\_\_\_\_.

BANCA EXAMINADORA

---

Prof. Me. Wellington Wagner Ferreira Sarmiento (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Me. Antônio de Lisboa Coutinho Júnior  
Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

---

Prof. Dr. Henrique Sérgio Lima Pequeno  
Universidade Federal do Ceará (UFC)

## **AGRADECIMENTOS**

Ao meu orientador, Prof. Me. Wellington Wagner Ferreira Sarmiento, que desde o primeiro momento confiou no meu potencial e realizou uma excelente orientação.

A minha família, que sempre me passou bons valores e é minha base de apoio para qualquer situação.

Aos colegas e professores do curso de Sistemas e Mídias Digitais, que contribuíram para a minha formação acadêmica.

## RESUMO

A utilização de Ambientes Virtuais de Aprendizagem (AVA), em cursos na modalidade de Educação a Distância, proporciona diferentes recursos de comunicação para a discussão entre alunos e professores, sendo os *web fóruns* uma ferramenta muito importante nesse processo. Observa-se nos cursos de licenciatura a distância em Matemática e Física ministrados pela Universidade Federal do Ceará (UFC) que se faz intenso uso de tais *web fóruns*, sendo um processo ainda problemático o uso de linguagem simbólica matemática e análise de gráficos nestes ambientes. Para solucionar parcialmente este problema, foi implementada uma ferramenta chamada *MathPost*, que permite a escrita e visualização de expressões matemáticas. No entanto ainda haveria o problema de geração e visualização de gráficos para ser resolvido. É neste cenário que se desenvolve o presente trabalho, onde foi criada a ferramenta *Gnuplot on Web* (GOW) que permite a utilização do *software Gnuplot* dentro do *web editor* de textos *CKEditor*, que pode ser utilizado em fóruns de Ambientes Virtuais de Aprendizagem. Os testes preliminares efetuados sobre essa aplicação possibilitaram a garantia de seu funcionamento tanto em situações esperadas, quanto no caso de exceções.

**Palavras-chave:** Plotagem de gráficos. Web Editor. Ambiente Virtual de Aprendizagem.

## ABSTRACT

The use of Virtual Learning Environments (VLE) in Distance Education courses provides different communication resources for discussion between students and teachers, and web forums are a very important tool in this process. It is observed in the distance degree courses in Mathematics and Physics taught by the Federal University of Ceará (UFC) that there is intense use of such web forums, with the use of mathematical symbolic language and graphic analysis in these environments being a still problematic process. To partially solve this problem, a tool called MathPost was implemented, which allows the writing and visualization of mathematical expressions. However, there would still be the problem of generating and viewing graphics to be solved. It is in this scenario that the present work is developed, where the Gnuplot on Web (GOW) tool was created, which allows the use of Gnuplot software within the CKEditor web text editor, which can be used in forums of Virtual Learning Environments. The preliminary tests carried out on this application made it possible to guarantee its operation both in expected situations and in the case of exceptions.

**Keywords:** Plotting Charts. Web Editor. Virtual learning environment.

## LISTA DE FIGURAS

Figura 1	– Exemplo de utilização do editor matemático MathPost .....	16
Figura 2	– Interface do Gnuplotter .....	17
Figura 3	– Arquitetura do GOW .....	18
Figura 4	– Lista de dependências da API do GOW .....	19
Figura 5	– Lista de dependências de desenvolvimento da API do GOW .....	20
Figura 6	– Estrutura de arquivos e diretórios do plug-in GOW .....	20
Figura 7	– Botão do GOW na barra de ferramentas do CKEditor .....	21
Figura 8	– Caixa de diálogo do plug-in do GOW no CKEditor .....	21
Figura 9	– Utilização da biblioteca comandante para execução do Gnuplot .....	23
Figura 10	– Montagem do script de plotagem do gráfico na API GOW .....	23
Figura 11	– Exemplo de plotagem bem sucedida .....	24
Figura 12	– Exemplo de plotagem mal sucedida .....	25
Figura 13	– Demonstração do menu de contexto do plug-in com a opção de editar .....	26
Figura 14	– Trecho da codificação dos testes .....	27
Figura 15	– Resultados dos testes da API do GOW com o Jest .....	28
Figura 16	– Resultado do teste de plotagem com o Selenium IDE .....	29
Figura 17	– Resultado do teste de edição de um gráfico com o Selenium IDE .....	29

## LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
AVA	Ambiente Virtual de Aprendizagem
CORS	<i>Cross-Origin Resource Sharing</i>
EAD	Educação a Distância
GOW	Gnuplot on Web
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hypertext Transfer Protocol Secure</i>
JSON	<i>JavaScript Object Notation</i>
PNG	<i>Portable Network Graphics</i>
REST	<i>Representational State Transfer</i>
SDK	<i>Software Development Kit</i>
SOAP	<i>Simple Object Access Protocol</i>
UAB	Universidade Aberta do Brasil
UFC	Universidade Federal do Ceará
URL	<i>Uniform Resource Locator</i>
WYSIWYG	<i>What You See Is What You Get</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	11
<b>1.1</b>	<b>Objetivos</b> .....	12
<i>1.1.1</i>	<i>Objetivo Geral</i> .....	12
<i>1.1.2</i>	<i>Objetivos Específicos</i> .....	12
<b>1.1</b>	<b>Estrutura do documento</b> .....	12
<b>2</b>	<b>REFERENCIAL TEÓRICO</b> .....	13
<b>2.1</b>	<b>Ambiente Virtual de Aprendizagem</b> .....	13
<b>2.2</b>	<b>Editor de texto CKEditor</b> .....	14
<b>2.3</b>	<b>Representational State Transfer</b> .....	14
<b>2.4</b>	<b>Trabalhos Relacionados</b> .....	15
<i>2.4.1</i>	<i>MathPost</i> .....	15
<i>2.4.2</i>	<i>Gnuplotter</i> .....	16
<b>3</b>	<b>METODOLOGIA</b> .....	17
<b>4</b>	<b>APLICAÇÃO GNUPLOT ON WEB</b> .....	18
<b>4.1</b>	<b>Arquitetura do Sistema</b> .....	18
<i>4.1.1</i>	<i>Back-end</i> .....	19
<i>4.1.2</i>	<i>Front-end</i> .....	20
<b>4.2</b>	<b>Aspectos funcionais</b> .....	21
<b>5</b>	<b>TESTES</b> .....	26
<b>5.1</b>	<b>Testes unitários realizados na API do GOW</b> .....	26
<b>5.2</b>	<b>Testes realizados no plug-in do GOW no CKEditor</b> .....	28
<b>5.3</b>	<b>Resultados</b> .....	28
<b>5.4</b>	<b>Análise dos Resultados</b> .....	29
<b>6</b>	<b>CONCLUSÃO</b> .....	29
	<b>REFERÊNCIAS</b> .....	31
	<b>APÊNDICE A – USER STORIES</b> .....	33

## 1 INTRODUÇÃO

Conforme observado por Malheiro et al. (2021, p. 143): “No Brasil, a educação a distância (EaD) ganhou amplitude no contexto do ensino superior ao longo das últimas décadas, tanto na oferta inicial, quanto na formação continuada”. Com o crescimento da Educação a Distância (EaD) e a consequente evolução dos Ambientes Virtuais de Aprendizagem (AVA), foi ampliada a gama de recursos digitais com fins educacionais, dentre esses recursos Pedro e Razera (2018, p. 470) destacam que “Os fóruns online constituem-se como uma das ferramentas educativas digitais mais utilizadas para a troca de informação e a construção do conhecimento entre pares em ambientes virtuais de aprendizagem (AVA)”.

A Universidade Federal do Ceará (UFC) conta com o Solar, um AVA desenvolvido pelo Instituto UFC Virtual para oferta de educação a distância, que dentre outros recursos possibilita discussões acadêmicas através de *web* fóruns. A ferramenta de fórum oferecida pelo Solar utiliza o *CKEditor*<sup>1</sup>, um editor de textos *Open Source* para a *Web* do tipo *What You See Is What You Get* (WYSIWYG), para a criação e edição de postagens, que permitem a inclusão de conteúdo na forma de texto e imagem.

Em parceria com a Universidade Aberta do Brasil (UAB) a UFC oferece cursos de graduação por meio de EAD. No curso de Licenciatura Plena em Matemática a carga horária está dividida em 80% virtual e 20% presencial, onde as aulas virtuais acontecem por meio do Solar.

Segundo Peixoto et al (2014), existe a necessidade do uso de recursos matemáticos nos fóruns de discussão em AVAs. Esta autora partiu de estudos dos ambientes Moodle, Claroline, EDX e Solar. O Solar foi analisado em termos de uso de fóruns de discussão nos cursos de Licenciatura em Matemática e Física a Distância, onde quando necessário inserir alguma equação ou exemplificar alguma fórmula matemática no fórum, era preciso utilizar editores externos e inserir como imagem na postagem. A proposta desta autora, foi implementar um editor para permitir a utilização de notação matemática dentro do fórum do Solar sem a necessidade de uma ferramenta externa.

Diante desse contexto é que se coloca o presente trabalho, onde se buscou ampliar a gama de recursos oferecidos aos alunos destes cursos à distância, possibilitando exibir graficamente funções descritas em fóruns para entendimento de um dado problema, através da criação de uma *interface* entre o editor de texto *CKEditor* utilizado no espaço de fórum do

---

<sup>1</sup> Disponível em <<https://ckeditor.com>>. Acesso em 03 fev. 2021.

Solar e o *software* de plotagem de gráficos *Gnuplot*<sup>2</sup>, um *software Open Source*, capaz de plotar gráficos por linha de comando a partir de dados e funções matemáticas, sendo também adotado por outros programas como mecanismo de plotagem. Como exemplo destes programas tem-se o *GNU Octave*<sup>3</sup>, *Maxima*<sup>4</sup> e pacote *gretl*<sup>5</sup>.

## 1.1 Objetivos

### 1.1.1 Objetivo Geral

Desenvolver uma ferramenta que possibilite a plotagem de gráficos a partir de funções e conjuntos de dados fornecidos.

### 1.1.2 Objetivos Específicos:

- a) Criar uma *Application Programming Interface* (API), que possibilite a comunicação com a ferramenta de geração de gráficos *Gnuplot*;
- b) Desenvolver um *plug-in* do *CKEditor* que possibilite a inserção e edição de gráficos na área do editor, por meio de uma *interface* com campos de entrada para os dados referentes ao gráfico desejado;
- c) Integrar a API proposta com o *plug-in* de inserção e edição de gráficos do *CKEditor*.

## 1.2 Estrutura do documento

Este trabalho está organizado em seis capítulos, incluindo este. O segundo capítulo reúne todo o arcabouço teórico necessário para o entendimento dos conceitos abordados. No terceiro capítulo é apresentada a metodologia utilizada na realização deste trabalho. Os aspectos técnicos e de implementação do produto, são explicados no quarto capítulo. Os testes e análise dos resultados são apresentados no quinto capítulo. E a conclusão no sexto capítulo com considerações finais e discussão sobre possíveis trabalhos futuros.

---

<sup>2</sup> Disponível em <<http://www.gnuplot.info/>>. Acesso em 03 fev. 2021.

<sup>3</sup> Disponível em <<https://www.gnu.org/software/octave/index>>. Acesso em 03 fev. 2021.

<sup>4</sup> Disponível em <<https://maxima.sourceforge.io/>>. Acesso em 03 fev. 2021.

<sup>5</sup> Disponível em <<http://gretl.sourceforge.net/>>. Acesso em 03 fev. 2021.

## 2 REFERENCIAL TEÓRICO

Nesta seção, os principais termos e conceitos abordados neste trabalho serão apresentados e explicados.

### 2.1 Ambiente Virtual de Aprendizagem

Salvador et al. (2017, p. 605) compreendem AVA como “um sistema computacional que integra funcionalidades e ferramentas as quais possibilitam a construção de um processo de ensino-aprendizagem interativo, on-line, acessado por navegadores na internet ou em redes locais”.

De acordo com Coelho e Tedesco (2017, p. 611):

O ambiente virtual de aprendizagem (AVA) é a parte mais visível de todo o arcabouço tecnológico utilizado para dar suporte à realização da EAD. É por meio dele que são disponibilizados os conteúdos dos cursos e são promovidas as interações entre os participantes.

Dias, Coelho e Brasileiro (2018, p. 144) afirmam que “Quando se fala em EAD, tem que levar em consideração que o AVA é ontológico, isso quer dizer que é imprescindível desenvolver um processo de aprendizagem onde o aprendiz interaja dentro do meio virtual. “.

Para Hessel e Oliveira (2017, p. 157), “Uma das principais vantagens de um ambiente virtual de aprendizagem é a de expandir as alternativas físicas. Outra possibilidade é que, no ambiente virtual a construção do conhecimento não precisa ficar restrita à transmissão unilateral de uma informação.”.

Sobre o AVA Solar, Coutinho, Moreira e Sarmiento (2013, p.488), afirmam que:

(...) o SOLAR foi desenvolvido potencializando o aprendizado a partir da relação com a própria interface gráfica do ambiente, sendo desenvolvido para que o usuário tenha rapidez no acesso as páginas e ao conteúdo, fácil navegabilidade e compatibilidade com navegadores. Nele, o interagente se sente seguro para explorar os espaços disponibilizados. O ambiente é apoiado numa filosofia de interação e não de controle.

Ao analisar e discutir sobre o recurso de fórum do Solar, Martins (2021, p. 46) afirma que, “é uma ferramenta pautada na construção individual e colaborativa do conhecimento, uma vez que a ferramenta promove debate e reflexão acerca do assunto

proposto.”, desta forma é possível perceber que se trata de uma ferramenta importante para as atividades de educação a distância e cujos recursos podem facilitar ou não a comunicação entre professor-aluno e aluno-aluno. Criar uma ferramenta que permita melhorar a postagem de dúvidas e discussão de problemas matemáticos ou que usem notação matemática, dentro do fórum, pode possibilitar o maior uso desta ferramenta junto aos cursos da área de ciências exatas, incluindo os que hoje são mantidos pela UFC.

## 2.2 Editor de texto CKEditor

O *CKEditor* é um editor *HyperText Markup Language* (HTML) WYSIWYG de código aberto, muito adotada pela comunidade mundo afora, que oferece diversas opções de configuração e personalização. Os recursos deste editor podem ser estendidos através de plugins desenvolvidos com seu *Software Development Kit* (SDK), pela comunidade. (CKEDITOR, 2021).

Esse editor é usado em vários dos recursos presentes no AVA Solar, incluindo seus *web fóruns*. Assim, a ferramenta a ser projetada deve integrável a este ambiente, sem, no entanto, criar uma relação de completa dependência com relação a este software. Por isso se escolheu usar integração de softwares com base em API. No caso específico deste trabalho, *API Restful*.

## 2.3 Representational State Transfer

Kurose e Ross (2013), ao tratarem sobre arquiteturas de aplicação de rede, afirmam que:

Em uma arquitetura cliente-servidor há um hospedeiro sempre em funcionamento, denominado servidor, que atende a requisições de muitos outros hospedeiros, denominados clientes. Um exemplo clássico é a aplicação Web na qual um servidor Web que está sempre em funcionamento atende a solicitações de navegadores de hospedeiros clientes.

Um *Web Service* é uma aplicação cliente-servidor baseada em HTTP/HTTPS<sup>6</sup>. Para o W3C (2004), um *Web Service* pode ser definido como o um sistema de software projetado de modo que ofereça um meio padrão de interoperabilidade entre diferentes aplicações em variadas estruturas e plataformas. Em um *Web Service* existe a expectativa

---

<sup>6</sup> Hypertext Transfer Protocol/ Hypertext Transfer Protocol Secure

compartilhada sobre o modo como o serviço se comporta, em particular nas respostas para as mensagens a ele enviadas.

Segundo Pereira et al. (2018), o *Representational State Transfer* (REST) e *Simple Object Access Protocol* (SOAP) são os principais modelos de *web services*. Conforme Wikipédia (2021), REST é um estilo de arquitetura de software que usa um subconjunto de *Hypertext Transfer Protocol* (HTTP). Os *web services* que estão em conformidade com a arquitetura REST, podem ser denominados RESTful. Diferente do modelo SOAP, para serviços baseados RESTful não existe um padrão oficial, isso se dá, pois, o REST é um estilo de arquitetura, enquanto o SOAP é um protocolo. Dentre os princípios REST estão ausência de estado, uso de um sistema de camadas e arquitetura cliente-servidor.

Conforme MDN (2021), muitas vezes APIs HTTP são chamadas de APIs RESTful ou serviços RESTful, contudo deve-se observar a adoção da conformidade com a arquitetura REST para tal.

## 2.4 Trabalhos Relacionados

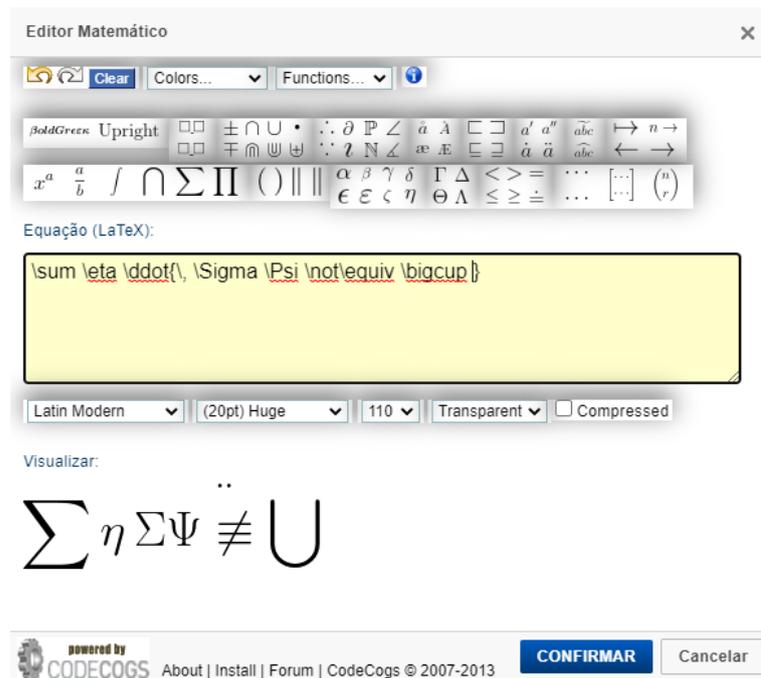
### 2.4.1 *MathPost*

O *MathPost* é um editor matemático, implementado no Solar 2.0 com base no *plug-in Equation Editor*, desenvolvido pela empresa *CodeCogs*<sup>7</sup> e adicionado na forma de *plug-in*, ao editor de textos do fórum de discussão do Solar o *CKEditor*. O editor oferece um menu de símbolos matemáticos, onde o usuário pode selecionar o símbolo desejado clicando sobre ele, uma vez selecionado o símbolo a caixa de edição carrega o respectivo conteúdo em formato *LaTeX*, permitindo que os valores sejam editados conforme a necessidade. Uma vez finalizada a escrita do conteúdo desejado, ele é exportado para o editor de texto do fórum em formato de imagem, podendo ainda retornar para a área do editor matemático caso tenha a necessidade (PEIXOTO, TORRES, *et al.*, 2014). A Figura 1 exemplifica a utilização da ferramenta.

---

<sup>7</sup> Disponível em: <<https://www.codecogs.com/latex/eqneditor.php?lang=pt-br>>. Acesso em 03 abr. 2021

Figura 1 - Exemplo de utilização do editor matemático MathPost



Fonte: Solar<sup>8</sup> (2021).

Como pôde-se perceber, nenhuma destas duas ferramentas permitem a plotagem e discussão sobre gráficos, assim, procurou-se desenvolver um *software* que resolvesse esse problema e trouxesse as vantagens destas ferramentas.

## 2.4.2 Gnuplotter

O *Gnuplotter*<sup>9</sup> é uma ferramenta *web*, que permite a execução de *scripts Gnuplot* no navegador com plotagem imediata do resultado na tela. A ferramenta foi desenvolvida a partir do projeto *gnuplot-JS*<sup>10</sup> que combina: o *emscripten*<sup>11</sup> uma solução que compila código C/C++ em *Javascript* com a ferramenta *Gnuplot*. Sua utilização requer conhecimento na linguagem *script* do *Gnuplot*. A *interface* da aplicação pode ser vista na Figura 2.

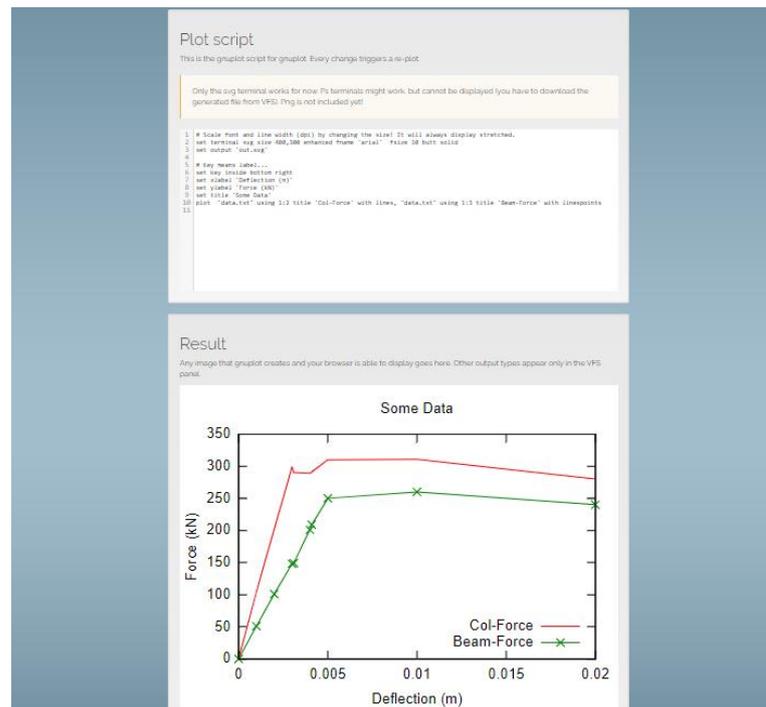
<sup>8</sup> Disponível em: <<https://solar.virtual.ufc.br>>. Acesso em 08 abr. 2021

<sup>9</sup> Disponível em: <<http://gnuplot.respawned.com/>>. Acesso em 08 abr. 2021

<sup>10</sup> Disponível em: <<https://github.com/chhu/gnuplot-JS/>>. Acesso em 08 abr. 2021

<sup>11</sup> Disponível em: <<https://emscripten.org/>>. Acesso em 07 abr. 2021

Figura 2 - Interface do Gnuplotter



Fonte: Gnuplotter (2021).

### 3 METODOLOGIA

O estudo inicialmente foi de caráter exploratório. Para Gil (2002), são classificadas como pesquisas exploratórias aquelas que almejam trazer mais familiaridade com o problema, constituindo hipóteses e objetivando torná-lo mais explícito. Nesta etapa foram analisados o funcionamento do fórum do Solar e as possibilidades de recursos a serem inseridos nele.

A segunda etapa envolveu o desenvolvimento da aplicação proposta, com a adoção do *Scrum*, devido ao curto prazo para finalização do trabalho. O *Scrum* é uma metodologia ágil utilizada na gestão e planejamento de projetos de *software* que se baseia em pequenos ciclos de execução de atividades chamados de *Sprints*. (DESENVOLVIMENTOÁGIL, 2014). A partir de *User Stories* foram definidas atividades e *sprints* de cinco dias para desenvolvimento de cada etapa do projeto. As *User Stories* estão presentes no apêndice A.

O GitHub<sup>12</sup>, por ser um serviço amplamente utilizado em projetos de *softwares open source*, foi escolhido para hospedagem pública e controle de versão do código fonte da

<sup>12</sup> Disponível em: <<https://github.com/rmatheuss/GnuPlotCkEditor>>. Acesso em 13 abr. 2021

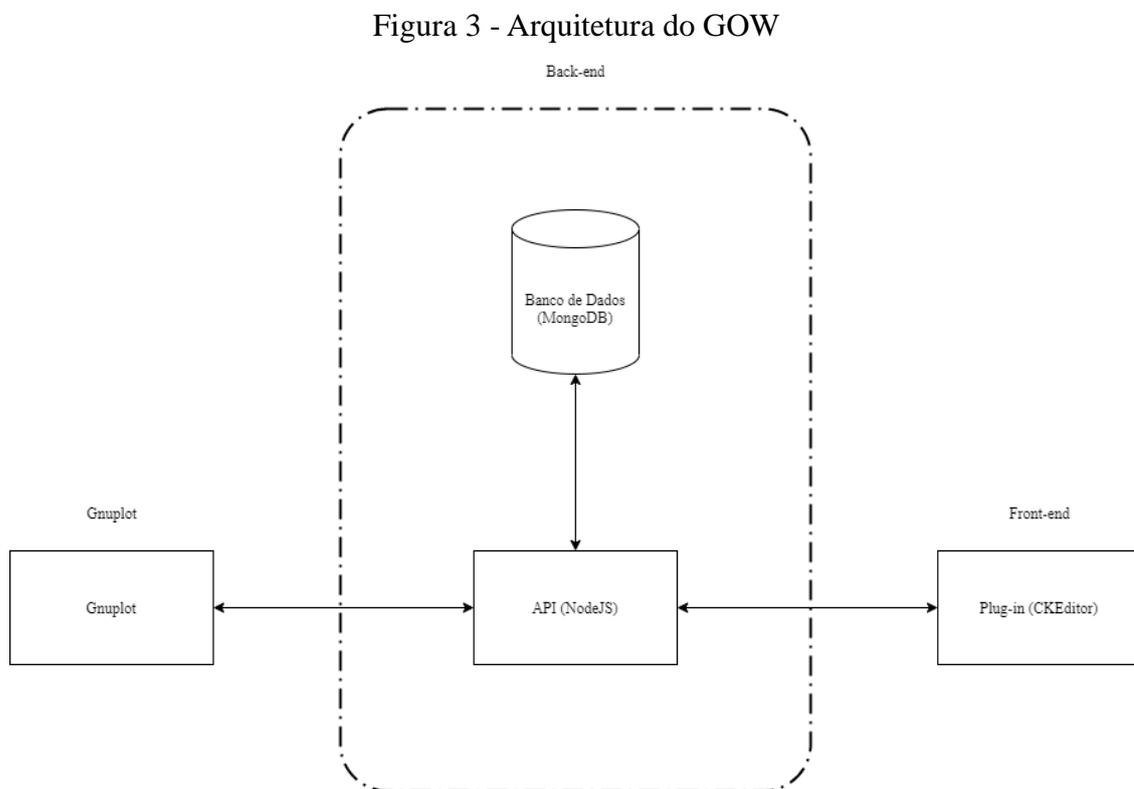
aplicação. Como última etapa da pesquisa foram realizados testes unitários para análise do desempenho da aplicação e testes funcionais com objetivo de refletir a ótica do usuário, observando os requisitos funcionais do *software* sem levar em conta os detalhes de operação interna. ( DEVMEDIA, 2021).

## 4 APLICAÇÃO GNUPLOT ON WEB

Esta seção descreve os aspectos técnicos de planejamento e desenvolvimento de todos os elementos que compõem o *Gnuplot on Web* (GOW).

### 4.1 Arquitetura do Sistema

O GOW é uma aplicação distribuída, baseada na arquitetura cliente-servidor, onde o *backend* foi construído na forma de uma API RESTful em Node.js com um banco de dados não relacional MongoDB para a persistência dos dados. O *frontend* foi desenvolvido na forma de um *plug-in* do *CKEditor* utilizando a linguagem *JavaScript*. A arquitetura proposta pode ser vista na Figura 3.



Fonte: Elaborado pelo Autor.

### 4.1.1 Back-end

A API RESTful do GOW foi implementada com a linguagem *Javascript* na plataforma *Node.js* pois ele é um ambiente *open source* e que permite a criação de aplicações do lado do servidor. O *framework Express.js* foi utilizado para facilitar a configuração da aplicação e o gerenciamento de *middlewares*, rotas, verbos, requisições, respostas e entrega de arquivos estáticos. Foi utilizado o gerenciador de pacotes *Yarn*, com o projeto contando com as seguintes dependências:

- a) **comandante**: Pacote responsável pela inicialização do *Gnuplot* via *Node.js*;
- b) **cors**: *Middleware* para configuração de *Cross-Origin Resource Sharing* (CORS);
- c) **dotenv**: Pacote para carregamento de variáveis de ambiente;
- d) **express**: *Framework* para aplicações web em *Node.js*;
- e) **mongoose**: Pacote para modelagem de objetos *MongoDB*;
- f) **morgan**: *Middleware* para log de chamadas HTTP.

Dentro de um projeto *Node.js* as dependências e suas respectivas versões são mantidas em um arquivo chamado *package.json*, como é possível ver na Figura 4:

Figura 4 – Lista de dependências da API do GOW

```
"dependencies": {  
  "comandante": "^0.0.1",  
  "cors": "^2.8.5",  
  "dotenv": "^8.2.0",  
  "express": "^4.17.1",  
  "mongoose": "^5.12.2",  
  "morgan": "^1.10.0"  
},
```

Fonte: Elaborado pelo Autor.

O projeto também conta com uma lista de dependências de desenvolvimento, que são ferramentas utilizadas para agilizar a codificação e os testes, não devendo ser utilizadas em um ambiente produtivo, a API do GOW conta com as seguintes dependências de desenvolvimento:

- g) **jest**: *Framework* para desenvolvimento de testes automatizados;
- h) **nodemon**: Biblioteca para reinício automático de aplicações;
- i) **supertest**: Biblioteca para teste de requisições HTTP.

As dependências de desenvolvimento podem ser vistas na Figura 5:

Figura 5 - Lista de dependências de desenvolvimento da API do GOW

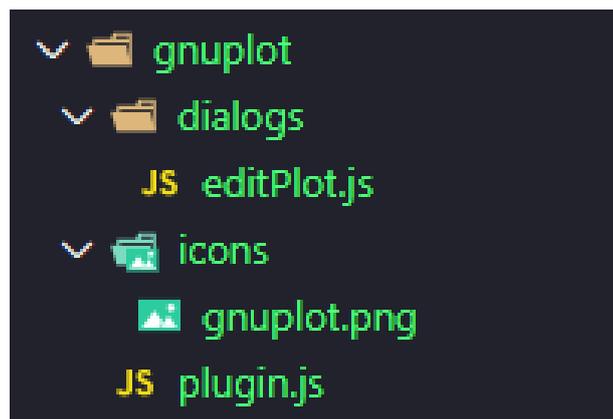
```
"devDependencies": {  
  "jest": "^26.6.3",  
  "nodemon": "^2.0.7",  
  "supertest": "^6.1.3"  
}
```

Fonte: Elaborado pelo Autor.

#### 4.1.2 Front-end

O *plug-in* do GOW foi desenvolvido na linguagem *JavaScript*, com a utilização do *Software Development Kit* (SDK) do editor de texto CKEditor.

Figura 6 - Estrutura de arquivos e diretórios do plug-in GOW



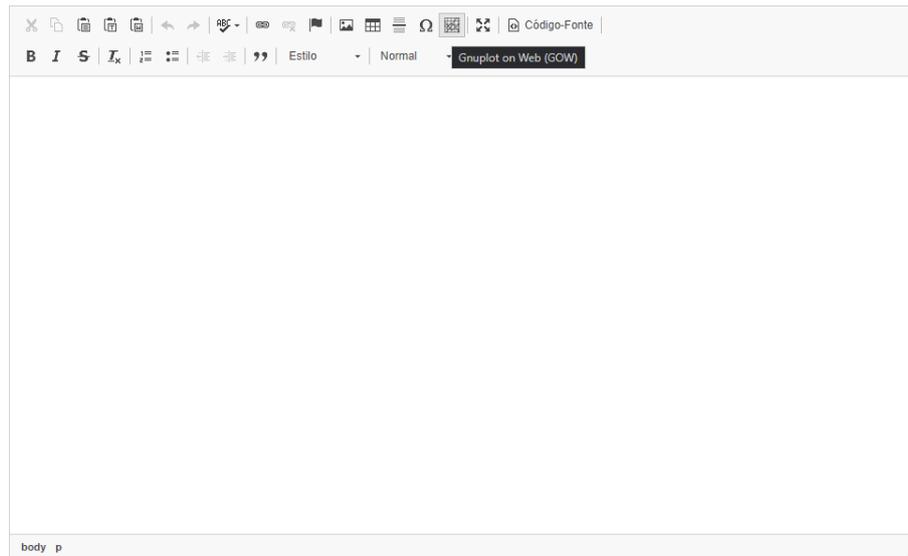
Fonte: Elaborado pelo Autor.

O arquivo *plugin.js* é o arquivo responsável pela lógica do *plug-in*, a pasta *icons* contém o arquivo *gnuplot.png* que é utilizado na barra de ferramentas do editor como ícone do *plug-in*, a pasta *dialogs* contém o arquivo *editPlot.js* que é responsável por definir a caixa de diálogo de criação e edição dos gráficos.

## 4.2 Aspectos Funcionais

O GOW é acionado através de um botão na barra de ferramentas do *CKEditor*. A Figura 7 mostra a barra de ferramentas do *CKEditor* com o botão do GOW.

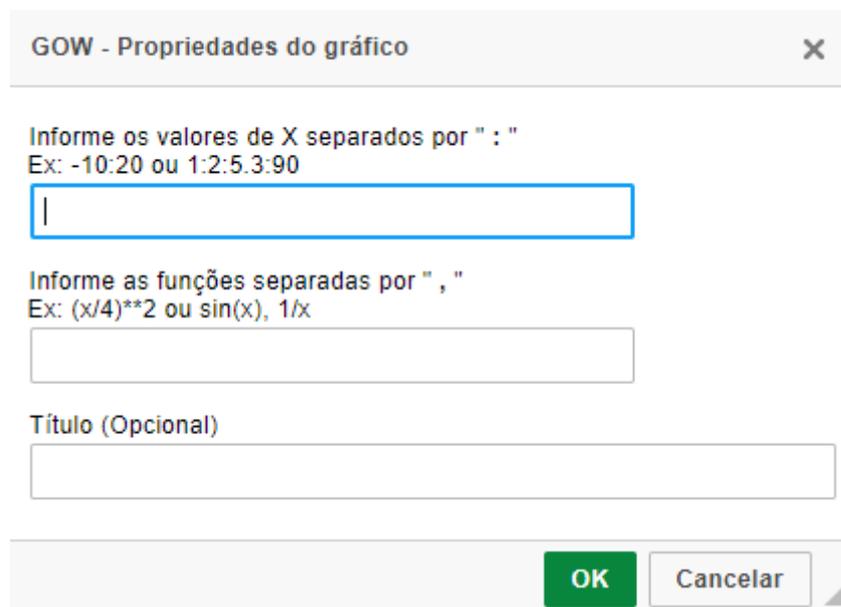
Figura 7 - Botão do GOW na barra de ferramentas do *CKEditor*



Fonte: Elaborado pelo Autor.

Ao clicar no botão é aberta a caixa de diálogo onde o usuário informará as propriedades do gráfico que deseja plotar, conforme demonstra a Figura 8.

Figura 8 - Caixa de diálogo do *plug-in* do GOW no CKEditor

A screenshot of a dialog box titled 'GOW - Propriedades do gráfico'. The dialog box has a close button (X) in the top right corner. It contains three text input fields. The first field is labeled 'Informe os valores de X separados por " : "' and has an example 'Ex: -10:20 ou 1:2:5.3:90'. The second field is labeled 'Informe as funções separadas por " , "' and has an example 'Ex: (x/4)\*\*2 ou sin(x), 1/x'. The third field is labeled 'Título (Opcional)'. At the bottom of the dialog box, there are two buttons: 'OK' and 'Cancelar'.

Fonte: Elaborado pelo Autor.

Ao informar as propriedades do gráfico nas caixas de texto e confirmando-se com o botão “OK”, é realizada pelo *plug-in* uma chamada para a API do GOW, passando as propriedades informadas pelo usuário, no corpo da requisição HTTP em formato *JavaScript Object Notation* (JSON).

A API ao receber essa requisição, captura as propriedades enviadas no corpo, em seguida gera-se uma sequência de caracteres aleatória através do utilitário *Crypto*<sup>13</sup> do *Node.js*. A sequência de caracteres gerada é utilizada como identificador único do gráfico.

O gráfico é plotado como um arquivo de imagem no formato *Portable Network Graphics* (PNG), nomeado com o identificador único gerado anteriormente e armazenado no servidor. O processo de plotagem acontece através da biblioteca *open source comandante*<sup>14</sup>, que executa o *Gnuplot* passando o *script* montado pela aplicação, a implementação com a utilização da biblioteca *comandante* foi baseada no projeto *open source node-gnuplot*<sup>15</sup>. A Figura 9 e a Figura 10, demonstram parte do código responsável pela plotagem do gráfico. No primeiro código é feita a *interface* com o *software Gnuplot*, já o segundo trata da montagem do *script* para ser executado no *Gnuplot*.

---

<sup>13</sup> Disponível em: <<https://nodejs.org/api/crypto.html>>. Acesso em 03 abr. 2021

<sup>14</sup> Disponível em: <<https://github.com/substack/comandante>>. Acesso em 03 abr. 2021

<sup>15</sup> Disponível em: <<https://github.com/davvo/node-gnuplot>>. Acesso em 03 abr. 2021

Figura 9 - Utilização da biblioteca *comandante* para execução do *Gnuplot*

```

plotar = () => {
  var run = require('comandante')
  var plot = run('gnuplot', []);

  plot.print = function (data, options) {
    plot.write(data);
    if (options && options.end) {
      plot.end();
    }
    return plot;
  };

  plot.println = function (data, options) {
    return plot.print(data + '\n', options);
  };

  ['set', 'unset', 'plot', 'splot', 'replot'].forEach(function (name) {
    plot[name] = function (data, options) {
      if (data) {
        return plot.println(name + ' ' + data, options);
      }
      return plot.println(name, options);
    };
  });
  return plot;
}

```

Fonte: Elaborado pelo Autor.

Figura 10 – Montagem do script de plotagem do gráfico na API GOW

```

routes.post("/plotGraph", async (req, res) => {
  const { pX, pY, pTitle, pFunctions, pKey } = req.body;

  let plotTitle = "GoW (Gnuplot on Web)";
  if (pTitle) {
    plotTitle = "GoW (Gnuplot on Web) - " + pTitle.replace("GoW (Gnuplot on Web) - ", "");
  }

  let fileExt = "png";
  let fileName = crypto.randomBytes(16).toString("hex");
  if (pKey) fileName = pKey;

  try {
    let errors = "";
    let p = plotar();

    if (pY) {
      p.set(`term ${fileExt}`)
      .set(`output "${process.env.APP_FILES_PATH}${fileName}.${fileExt}"`)
      .set(`title "${plotTitle}"`)
      .set(`xrange ${pX}`)
      .set(`yrange ${pY}`)
      .plot(pFunctions)
      .end()
    }
    else{
      p.set(`term ${fileExt}`)
      .set(`output "${process.env.APP_FILES_PATH}${fileName}.${fileExt}"`)
      .set(`title "${plotTitle}"`)
      .set(`xrange ${pX}`)
      .plot(pFunctions)
      .end()
    }
  }

  p.stderr.on('data', (data) => { errors += `${data}`; });

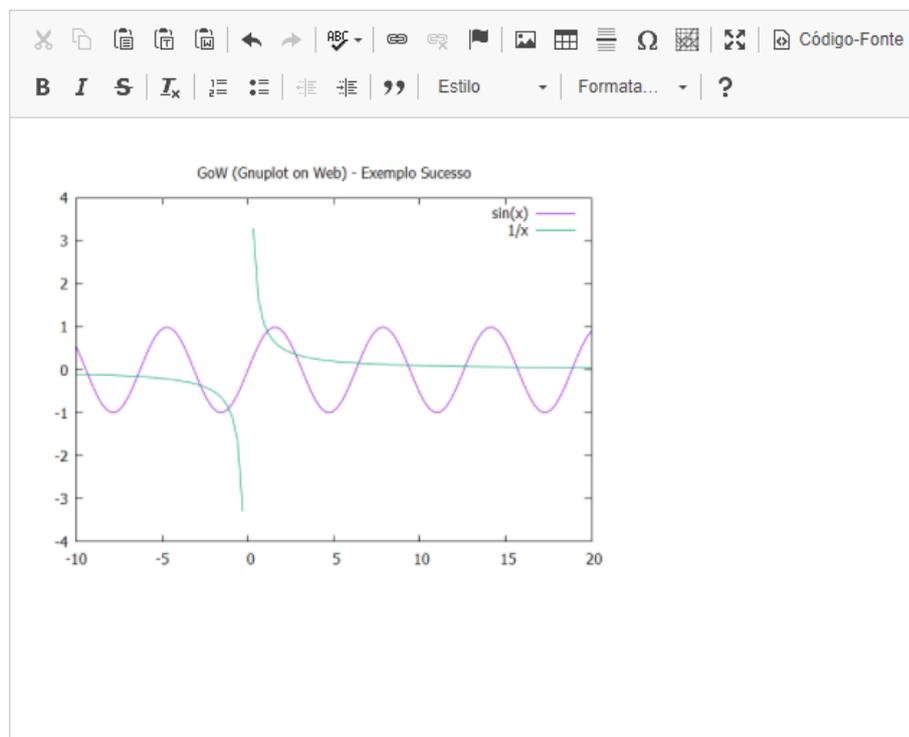
  p.on('exit', async (code) => {
    if (errors) {

```

Fonte: Elaborado pelo Autor.

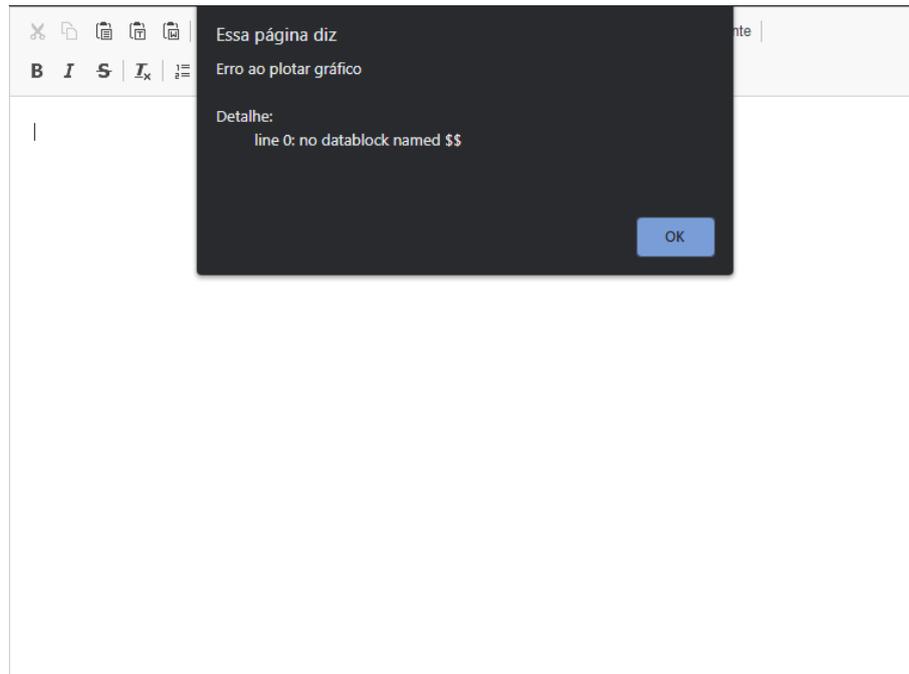
Após a plotagem do gráfico em forma de arquivo, são persistidas no *MongoDB*, todas as informações recebidas no corpo da requisição pela API: o título, o identificador único do gráfico e a *Uniform Resource Locator* (URL) de acesso a imagem. Estas informações, são enviadas para o *plug-in* do GOW no *CKEditor*, como resposta da requisição em formato JSON com o código de status HTTP 200 OK, que indica que a requisição foi processada com sucesso. Caso ocorra algum erro ou exceção na API, será enviado como resposta o erro e seu detalhamento em formato JSON, com o código de status HTTP 500 *Internal Server Error*, que indica que houve um erro no servidor ao processar a requisição. A Figura 11 mostra um caso em que a plotagem foi realizada com sucesso no CKEditor. A Figura 12 representa um caso onde ocorreu um erro ao processar a plotagem solicitada, em virtude de ter sido escrita erroneamente uma função.

Figura 11 - Exemplo de plotagem bem sucedida



Fonte: Elaborado pelo Autor.

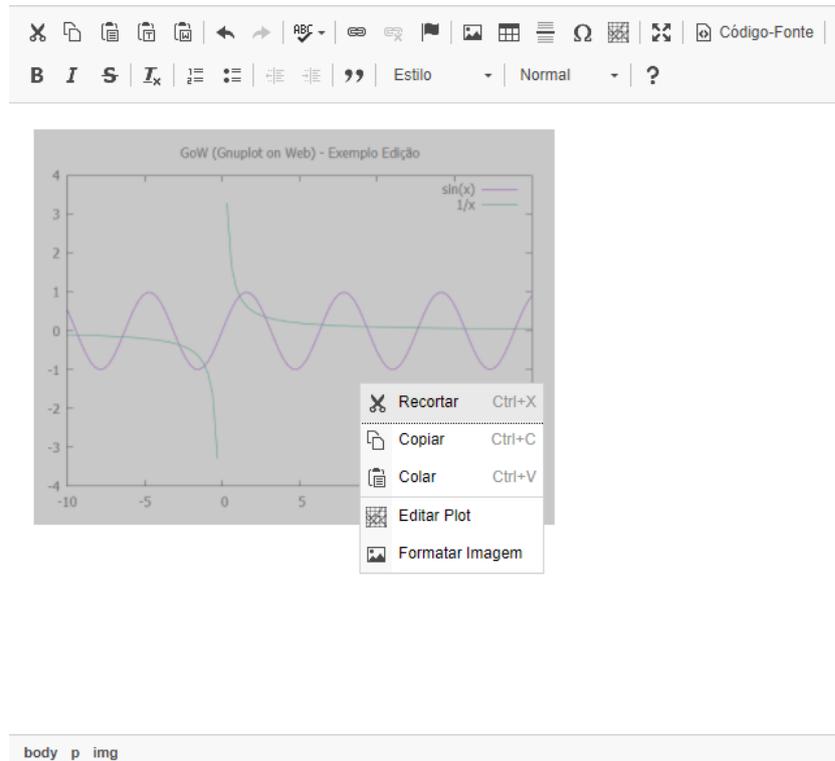
Figura 12 - Exemplo de plotagem mal sucedida



Fonte: Elaborado pelo Autor.

As imagens dos gráficos plotados contém as informações necessárias para sua edição, que são mantidas no atributo *alt* da *tag img* do HTML. A edição do gráfico é feita através de um clique na imagem com o botão direito do *mouse*, que exibirá um menu de contexto com o botão “Editar Plot”. A Figura 13 mostra o menu de contexto do *plug-in*.

Figura 13 - Demonstração do menu de contexto do *plug-in* com a opção de editar



## 5 TESTES

Nesta seção são detalhados os testes realizados e discutidos os resultados obtidos. Os testes feitos foram os unitários e o funcional.

### 5.1 Testes unitários realizados na API do GOW

Para a implementação dos testes da API do GOW, foi utilizado o *framework open source Jest*. Este *framework* desenvolvido pela equipe do *Facebook*, requer pouca configuração, possui vasta documentação e oferece uma grande quantidade de recursos para a escrita de testes de forma rápida e simples para códigos em *Javascript* ( JEST, 2021). Também foi utilizado o módulo *supertest*, juntamente com o Jest, para conexão com a API.

Os testes unitários da aplicação foram baseados em seis cenários de teste. A Figura 14 mostra um trecho de exemplo dos testes implementados.

Figura 14 - Trecho da codificação dos testes

```

const request = require('supertest');
const app = require('../src/app');

describe('Plotagem', () => {
  it('retorna um JSON com informações do gráfico plotado, uma vez que sejam passados parâmetros válidos de função e válidos de intervalo de X',
    async () => {
      const param = {
        "pKey": '',
        "pX": '[2:10]',
        "pTitle": "Teste - Cenário 1",
        "pFunctions": "sin(x), 1/x",
      };

      const response = await request(app)
        .post('/plotGraph')
        .send(param)
        .set('Accept', 'application/json');

      expect(response.status).toBe(200);
    });

  it('retorna um JSON com informações de erro, uma vez que sejam passados parâmetros inválidos de função e válidos de intervalo de X',
    async () => {
      const param = {
        "pKey": '',
        "pX": '[2:10]',
        "pTitle": "Teste - Cenário 2",
        "pFunctions": "$$",
      };

      const response = await request(app)
        .post('/plotGraph')
        .send(param)
        .set('Accept', 'application/json');

      expect(response.status).toBe(500);
    });
});

```

Fonte: Elaborado pelo Autor.

Foram utilizados os seguintes cenários para a implementação dos testes:

- a) A aplicação deve responder a requisição, com um JSON com informações do gráfico plotado, uma vez que sejam passados parâmetros válidos de função e válidos de intervalo de X (eixo);
- b) A aplicação deve responder a requisição, com um JSON com informações de erro, uma vez que sejam passados parâmetros inválidos de função e válidos de intervalo de X (eixo);
- c) A aplicação deve responder a requisição, com um JSON com informações de erro, uma vez que sejam passados parâmetros válidos de função e inválidos de intervalo de X (eixo);
- d) A aplicação deve responder a requisição, com um JSON com informações de erro, uma vez que sejam passados parâmetros inválidos de função e inválidos de intervalo de X (eixo);
- e) A aplicação deve responder a requisição, com um JSON com informações de erro, uma vez que não sejam passados parâmetros de função e de intervalo de X (eixo);

- f) A aplicação deve responder a requisição, com um JSON com informações do gráfico editado, uma vez que sejam passados parâmetros válidos de função, válidos de intervalo de X (eixo) e uma chave de gráfico plotado anteriormente.

## 5.2 Testes realizados no *plug-in* do GOW no CKEditor

Para a implementação no *plug-in* do GOW no CKEditor, foi utilizada a ferramenta *open source Selenium IDE*<sup>16</sup> de fácil configuração e utilização, que permite a realização de testes funcionais em aplicativos web.

Foi utilizado o recurso de gravação de ações do *Selenium IDE*, a partir disso foi realizado o processo de plotagem de um gráfico, com todos os eventos de interação com a *interface* do *plug-in* do GOW sendo capturados, o mesmo foi repetido para o processo de edição de um gráfico plotado. Ao fim da gravação da sequência de eventos de interação dos processos de plotagem e edição, essas interações foram estressadas de forma automática pelo *Selenium IDE*.

## 5.3 Resultados

Os testes da API, dos seis cenários levantados foram executados pelo *Jest*, em aproximadamente 4,9 segundos, obtendo sucesso em todos. Os resultados obtidos podem ser vistos na Figura 15.

Figura 15 - Resultados dos testes da API do GOW com o *Jest*

```

PASS  __tests__/integration/plot.test.js
Plotagem
  ✓ retorna um JSON com informações do gráfico plotado, uma vez que sejam passados parâmetros válidos de função e válidos de intervalo de X (999 ms)
  ✓ retorna um JSON com informações de erro, uma vez que sejam passados parâmetros inválidos de função e válidos de intervalo de X (159 ms)
  ✓ retorna um JSON com informações de erro, uma vez que sejam passados parâmetros válidos de função e inválidos de intervalo de X (275 ms)
  ✓ retorna um JSON com informações de erro, uma vez que sejam passados parâmetros inválidos de função e inválidos de intervalo de X (158 ms)
  ✓ retorna um JSON com informações de erro, uma vez que não sejam passados parâmetros de função e de intervalo de X (236 ms)
  ✓ retorna um JSON com informações do gráfico editado, uma vez que sejam passados parâmetros válidos de função, válidos de intervalo de X e uma chave de gráfico plotado anteriormente (335 ms)

Test Suites: 1 passed, 1 total
Tests:       6 passed, 6 total
Snapshots:  0 total
Time:        4.952 s, estimated 8 s
Ran all test suites.

```

Fonte: Elaborado pelo Autor.

Os testes realizados pelo Selenium IDE da interface do *plug-in* do CKEditor, foram bem sucedidos, os resultados obtidos com o teste da plotagem de um novo gráfico

<sup>16</sup> Disponível em: <<https://www.selenium.dev/selenium-ide/>>. Acesso em 13 abr. 2021

podem ser vistos na Figura 16, e os resultados obtidos com o teste de edição de um gráfico plotado podem ser vistos na Figura 17.

Figura 16 - Resultado do teste de plotagem com o Selenium IDE

```
Running 'Plotar gráfico' 02:11:42
1. open on file:///D:/Projetos/GnuPlotCkEditor/ckeditor/samples/index.html OK 02:11:43
'Plotar gráfico' completed successfully 02:11:43
```

Fonte: Elaborado pelo Autor.

Figura 17 - Resultado do teste de edição de um gráfico com o Selenium IDE

```
Running 'Editar gráfico' 02:14:15
1. open on file:///D:/Projetos/GnuPlotCkEditor/ckeditor/samples/index.html OK 02:14:15
'Editar gráfico' completed successfully 02:14:15
```

Fonte: Elaborado pelo Autor.

## 5.4 Análise dos Resultados

Ao analisar os resultados dos testes da API do GOW, é possível constatar que foi atingido um nível satisfatório de conformidade, entre os resultados esperados e os resultados obtidos, para cada um dos seis cenários de testes propostos os critérios pré-estabelecidos de aceitação foram cumpridos.

Nos testes funcionais realizados com a interface do *plug-in* do GOW no *CKEditor*, foi observado um comportamento dentro do esperado para a utilização, atingindo os requisitos funcionais, sem quebras na interface. Em situações de erro ou exceção, a devida mensagem de retorno é exibida na interface para o usuário.

## 6 CONCLUSÃO

Este trabalho objetiva o desenvolvimento de uma ferramenta para a plotagem de gráficos a partir de funções e conjuntos de dados fornecidos, que pudesse ser utilizada dentro de fóruns de discussão para ensino de matemática. Para o desenvolvimento da ferramenta proposta foi criada uma API de integração com o *Gnuplot*, bem como uma interface com o usuário na forma de um *plug-in* do *CKEditor*, que é o editor de textos utilizado nos fóruns do ambiente virtual Solar, até o momento da presente escrita. Também foi desenvolvida a

integração entre a API e o *plug-in*, com o objetivo de operacionalizar a plotagem dos gráficos solicitados pelo utilizador.

Além do desenvolvimento da aplicação, foram realizados testes funcionais centrados na *interface* a fim de validar a conformidade com os requisitos de funcionamento e testes unitários centrados na API a fim de validar as funções implementadas em termos de seu correto funcionamento e tratamento de exceções como: no caso de escrita de intervalos numéricos de entradas inválidas e mal formação na escrita das funções matemáticas a serem utilizadas. Em ambos os testes a ferramenta foi satisfatória.

Como trabalhos futuros, pretende-se integrar a aplicação aos fóruns do Solar e realizar testes com seus usuários a fim de determinar o grau de satisfação destes com o novo recurso. Além disso, serão realizadas melhorias na usabilidade e o tratamento de erros da aplicação.

## REFERÊNCIAS

CKEDITOR. **CKEditor 4**. Disponível em: <<https://ckeditor.com/ckeditor-4/>>. Acesso em: 23 fev. 2021.

COELHO, W. G.; TEDESCO, P. C. D. A. R. A percepção do outro no ambiente virtual de aprendizagem: presença social e suas implicações para Educação a Distância. **Revista Brasileira de Educação**, Rio de Janeiro, v. 22, n. 70, p. 609-624, Julho 2017. ISSN 1809-449X. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1413-24782017000300609&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-24782017000300609&lng=en&nrm=iso)>. Acesso em: 10 mar. 2021.

COUTINHO, E. F.; MOREIRA, L. O.; SARMENTO, W. W. F. MAAT - Sistema de Avaliação de Alunos e Tutores para um Ambiente Virtual de Aprendizagem. **Anais do Simpósio Brasileiro de Sistemas de Informação (SBSI)**, Porto Alegre, p. 487-498, 2013. Disponível em: <<https://sol.sbc.org.br/index.php/sbsi/article/view/5714>>. Acesso em: 15 fev. 2021.

DESENVOLVIMENTOÁGIL. **Scrum**. Disponível em: <<https://www.desenvolvimentoagil.com.br/scrum/>>. Acesso em: 07 maio 2021.

DEVMEDIA. **Testes funcionais de software**. Disponível em: <<https://www.devmedia.com.br/testes-funcionais-de-software/23565>>. Acesso em: 07 maio 2021.

DIAS, A. J. V. L.; COELHO, A. L.; BRASILEIRO, T. S. A. O Ambiente Virtual de Aprendizagem (AVA) e os desafios da gestão a distancia. **RECH-Revista Ensino de Ciências e Humanidades –Cidadania, Diversidade e Bem Estar**, [S. l.], v. 1, n. 1, p. 134-149, ago. 2018. ISSN 2594-8806.

GIL, A. C. **Como Elaborar Projetos de Pesquisa**. 4. ed. São Paulo: EDITORA ATLAS S.A., 2002.

HESSEL, A. M. D. G.; OLIVEIRA, W. C. D. O pensamento complexo aplicado no ambiente virtual de aprendizagem do SESC SP. **Laplage em Revista**, [S. l.], v. 3, n. 2, p. 150-158, 2017. ISSN 2446-6220. Disponível em: <<https://laplageemrevista.editorialaar.com/index.php/lpg1/article/view/311>>. Acesso em: 09 mar. 2021.

JEST. **JEST**. Disponível em: <<https://jestjs.io/pt-BR/>>. Acesso em: 06 abr. 2021.

JUNIOR, A. R. M. S. ARQUITETURA REST API E DESENVOLVIMENTO DE UMA APLICAÇÃO WEB SERVICE. **REVISTA ACADÊMICA ALCIDES MAYA-RAAM**, [S. l.], v. 1, n. 1, p. 1-59, 2019. Disponível em: <<http://raam.alcidesmaya.com.br/index.php/projetos/article/view/97>>. Acesso em: 22 mar. 2021.

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a Internet: uma abordagem top-down**. 6. ed. São Paulo: Pearson Education do Brasil Ltda., 2013.

MALHEIRO, C. A. L.; LIMA, V. S.; FERREIRA, R. D. D. S. Acessibilidade no Moodle: contribuições para inclusão dos estudantes com deficiência visual no ensino superior. **Laplage em Revista**, [S. l.], v. 7, n. 1, p. 142-154, Janeiro 2021. ISSN 2446-6220. Disponível em: <<https://laplageemrevista.editorialaar.com/index.php/lpg1/article/view/143>>. Acesso em: 04 mar. 2021.

MARTINS, B. S. B. Percepção dos alunos sobre as ferramentas do AVA Solar 2.0 no processo de aprendizagem, 2021. Disponível em: <<http://hdl.handle.net/10451/47107>>. Acesso em: 01 abr. 2021.

MDN. **REST**. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Glossary/REST>>. Acesso em: 02 abr. 2021.

PEDRO, N.; RAZERA, F. A INFLUÊNCIA DO PAPEL DO TUTOR NA INTERAÇÃO EM FÓRUMS DE DISCUSSÃO: UM ESTUDO EM EAD COM BASE NA ANÁLISE DE REDES SOCIAIS. **Revista e-Curriculum**, São Paulo, v. 16, n. 2, p. 467-493, 2018. ISSN 1809-3876. Disponível em: <<https://revistas.pucsp.br/index.php/curriculum/article/view/33932>>. Acesso em: 17 fev. 2021.

PEIXOTO, J. P. M. et al. MathPost: A integração de uma ferramenta gráfica utilizando LaTeX para discussões em webforum, Juiz de Fora, 2014. Disponível em: <[https://www.researchgate.net/publication/266230735\\_Mathpost\\_Integrating\\_a\\_Graphical\\_Tool\\_Using\\_Latex\\_for\\_Discussions\\_in\\_Webforums](https://www.researchgate.net/publication/266230735_Mathpost_Integrating_a_Graphical_Tool_Using_Latex_for_Discussions_in_Webforums)>. Acesso em: 12 nov. 2020.

PEREIRA, A. L. K. et al. ABORDAGEM RESTFUL EM SERVIÇOS WEB. **Revista Gestão em Foco**, n. 10, p. 460-465, 2018.

SALVADORI, P. T. C. D. O. et al. Objeto e ambiente virtual de aprendizagem: análise de conceito. **Revista Brasileira de Enfermagem**, Brasília, v. 70, n. 3, p. 572-579, Junho 2017. ISSN 0034-7167. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0034-71672017000300572&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0034-71672017000300572&lng=en&nrm=iso)>. Acesso em: 15 fev. 2021.

W3C. **W3C**, 2004. Disponível em: <<https://www.w3.org/TR/ws-arch/>>. Acesso em: 02 abr. 2021.

WIKIPÉDIA. **Representational state transfer**. Disponível em: <[https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)>. Acesso em: 22 mar. 2021.

## APÊNDICE A – USER STORIES

Como participante de um fórum, devo conseguir plotar um gráfico a partir de funções e de um intervalo de valores de X (eixo) passados por mim;

Como participante de um fórum, devo conseguir editar um gráfico já plotado por mim em uma postagem, para corrigir algum erro no gráfico;