



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

JONATHA KENNEDY MONTEIRO DA COSTA

**NODEGIS3D: SIMPLIFICANDO O DESENVOLVIMENTO E
EXPLORAÇÃO DE APLICAÇÕES WEB GIS EM 3D**

CAMPINA GRANDE - PB

2024

JONATHA KENNEDY MONTEIRO DA COSTA

**NODEGIS3D: SIMPLIFICANDO O DESENVOLVIMENTO E
EXPLORAÇÃO DE APLICAÇÕES WEB GIS EM 3D**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

Orientador: Professor Dr. Cláudio de Souza Baptista

CAMPINA GRANDE - PB

2024

JONATHA KENNEDY MONTEIRO DA COSTA

**NODEGIS3D: SIMPLIFICANDO O DESENVOLVIMENTO E
EXPLORAÇÃO DE APLICAÇÕES WEB GIS EM 3D**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

BANCA EXAMINADORA:

Cláudio de Souza Baptista

Orientador – UASC/CEEI/UFCG

Maxwell Guimarães de Oliveira

Examinador – UASC/CEEI/UFCG

Francisco Vilar Brasileiro

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 15 de maio de 2024.

CAMPINA GRANDE - PB

RESUMO

Atualmente, com o crescente impacto da informação geoespacial em nosso cotidiano, foram surgindo diversas ferramentas para o desenvolvimento de aplicações web de Sistemas de Informações Geográficas (SIG), dentre elas a NodeGIS. O NodeGIS3D representa uma evolução ao NodeGIS, adicionando funcionalidades de visualização e análise de dados geoespaciais em três dimensões, sem a necessidade de uma configuração complexa. O NodeGIS3D faz uso de uma integração de contêineres Docker, dessa forma simplificando o processo de deployment, possui também uma interface intuitiva, que permite uma interação fácil com as funcionalidades do sistema. A solução oferece ferramentas de consulta espacial, adição de recursos 3D e um ambiente tridimensional, permitindo uma navegação mais imersiva ao mapa. O NodeGIS3D também pode ser utilizado no ensino e aprendizagem de SIG, oferecendo uma aplicação robusta que pode ser inicializada facilmente.

NODEGIS3D: SIMPLIFYING THE DEVELOPMENT AND EXPLORATION OF 3D WEB GIS APPLICATIONS

ABSTRACT

Currently, with the increasing impact of geospatial information in our daily lives, various tools have emerged for developing web applications for Geographic Information Systems (GIS), including NodeGIS. NodeGIS3D represents an evolution of NodeGIS, adding functionalities for visualizing and analyzing geospatial data in three dimensions, without the need for complex configuration. NodeGIS3D makes use of Docker container integration, thus simplifying the deployment process. It also features an intuitive interface that allows easy interaction with the system's functionalities. The solution offers spatial query tools, addition of 3D features, and a three-dimensional environment, enabling more immersive navigation of the map. NodeGIS3D can also be used in GIS teaching and learning, providing a robust application that can be easily launched.

NodeGIS3D: Simplificando o Desenvolvimento e Exploração de Aplicações Web GIS em 3D

Jonatha Kennedy Monteiro da Costa

Laboratório de Sistemas da Informação
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil

jonatha.kennedy.costa@ccc.ufcg.edu.br

Cláudio de Souza Baptista

Laboratório de Sistemas da Informação
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil

baptista@computacao.ufcg.edu.br

RESUMO

Atualmente, com o crescente impacto da informação geoespacial em nosso cotidiano, foram surgindo diversas ferramentas para o desenvolvimento de aplicações web de Sistemas de Informações Geográficas (SIG), dentre elas a NodeGIS. O NodeGIS3D representa uma evolução ao NodeGIS, adicionando funcionalidades de visualização e análise de dados geoespaciais em três dimensões, sem a necessidade de uma configuração complexa. O NodeGIS3D faz uso de uma integração de contêineres Docker, dessa forma simplificando o processo de deployment, possui também uma interface intuitiva, que permite uma interação fácil com as funcionalidades do sistema. A solução oferece ferramentas de consulta espacial, adição de recursos 3D e um ambiente tridimensional, permitindo uma navegação mais imersiva ao mapa. O NodeGIS3D também pode ser utilizado no ensino e aprendizagem de SIG, oferecendo uma aplicação robusta que pode ser inicializada facilmente.

Palavras-chave

GIS, 3D GIS, web GIS.

Repositório

<https://github.com/jonathakmdc/node-gis3d>

1. INTRODUÇÃO

No contexto atual, as informações geoespaciais permeiam todos os setores da sociedade, desde a busca por restaurantes até a análise aprofundada de tendências climáticas. Cada vez mais, essas informações estão moldando o nosso cotidiano. Nesse cenário, os Sistemas de Informações Geográficas, ou *Geographic Information Systems* (GIS) em inglês, consolidam-se como ferramentas essenciais no processamento de informações, transformando dados em mapas interativos e ferramentas poderosas na tomada de decisões.

Grandes empresas como Microsoft¹, Google² e ESRI³ oferecem soluções robustas para o desenvolvimento de aplicações Web GIS (Sistema de Informações Geográficas web), e diversos sistemas de

gerenciamento de bancos de dados como PostgreSQL/PostGIS, MySQL e Oracle oferecem suporte à dimensão espacial.

No entanto, a criação de mapas num ambiente web pode se tornar um processo complexo e desafiador, demandando conhecimentos técnicos específicos. Foi nesse contexto que surgiu o NodeGIS [1], com o objetivo de simplificar e democratizar o desenvolvimento de aplicações Web GIS. É nesse mesmo contexto que surge o NodeGIS3D, solução aqui proposta.

Porém, como o nome já sugere, o NodeGIS3D tem um outro objetivo além da simplificação: a inclusão de funcionalidades para três dimensões (3D), de maneira que os dados possam ser visualizados em 3D, proporcionando uma imersão ainda maior.

O NodeGIS3D é uma ferramenta de código aberto que visa complementar o NodeGIS. Sua principal característica é a possibilidade de exibir mapas em 3D graças à integração com a biblioteca `deck.gl`⁴. Essa transição da visualização possibilita a criação de camadas de visualização complexas e altamente personalizáveis, abrindo novas possibilidades para a aplicação de GIS. É importante ressaltar a importância do suporte ao ambiente 3D em aplicações Web GIS, visto que essa funcionalidade supera as limitações associadas à visualização bidimensional. Com a capacidade de representar dados geoespaciais em 3D, torna-se viável comparar e analisar elementos como topografia e diferenças de elevação, o que enriquece as análises realizadas com uma perspectiva mais abrangente e detalhada.

Além de simplificar o desenvolvimento de aplicações Web GIS, o NodeGIS3D também pode desempenhar um papel significativo no ensino e aprendizagem de geoprocessamento. Ao oferecer uma ferramenta com visualizações interessantes de mapas no formato 3D, o NodeGIS3D pode auxiliar professores e alunos a explorarem conceitos de SIG de forma mais intuitiva.

Nas seções seguintes deste artigo, são explorados trabalhos relacionados (seção 2), a arquitetura (seção 3) e o processo de criação de uma aplicação Web GIS com o NodeGIS3D (seção 4). Em seguida, será abordada a avaliação da solução proposta (seção 5) e as considerações finais (seção 6).

2. TRABALHOS RELACIONADOS

A necessidade por soluções de geoprocessamento tem impulsionado o estudo e desenvolvimento de SIG, a gama de aplicações disponíveis atualmente oferece aos usuários interfaces

¹ <https://azure.microsoft.com/en-us/services/azure-maps/>

² <https://developers.google.com/maps>

³ <https://developers.arcgis.com/>

⁴ <http://deck.gl/>

interativas e experiências imersivas para a visualização e análise de dados geográficos, proporcionando um melhor entendimento do ambiente de estudo.

Na atual seção apresenta-se uma seleção de trabalhos relevantes para o desenvolvimento desse artigo, abrangendo diferentes estudos e tecnologias relacionadas ao desenvolvimento de aplicações Web GIS em 3D.

Qiu e Chen (2018) abordam o uso da tecnologia WebGL para apresentar gráficos 3D acelerados em páginas web. Eles destacam a importância da projeção de Mercator na visualização de mapas e descrevem como essa projeção é fundamental para representar coordenadas esféricas em um plano bidimensional, além disso, o artigo aborda a importância de arquivos B3DM na adição de camadas com alturas e informações e também destaca a utilização de tecnologias de código aberto para realizar experimentos relevantes na área. Já Mete et al. (2018) abordam a importância de bibliotecas de código aberto no desenvolvimento de aplicações web 3D. Os autores destacam a possibilidade de visualizar dados vetoriais e *raster* em diferentes escalas, possibilitando que os usuários consigam acessar e interagir com informações geoespaciais. A aplicação em questão utiliza a biblioteca Cesium e demonstra o potencial das tecnologias web para apresentar dados geoespaciais de maneira interativa e acessível.

Miao et al. (2017) discutem a importância do HTML5 e do WebGL ao apontar como essas tecnologias revolucionaram a forma como ferramentas Web GIS 3D são implementadas sem a necessidade de plugins adicionais e com suporte a aceleração por GPU, tornando a experiência mais rica, interativa e, principalmente, acessível pelo fato dessas tecnologias serem suportadas pelos principais navegadores atuais, garantindo uma ampla compatibilidade e acessibilidade para os usuários. Gesquière e Manin (2012) também apontam a importância do HTML5 e da WebGL para a utilização de outras ferramentas, sendo o CityGML o objeto de estudo, para a representação e troca de dados urbanos, ao permitir, na ferramenta, aglomerar dados, incluindo outras informações geométricas e dados semânticos por meio de arquivos XML.

Christen et al. (2012) apresentam seu projeto The OpenWebGlobe Project, um *framework* de código aberto, baseado em HTML5 e WebGL, que permite a criação de ambientes virtuais 3D em grande escala com conteúdo detalhado, seguindo na mesma linha Evangelidis et al. (2018) abordam o uso de tecnologias WebGL mas com o diferencial sendo para a criação de animações e efeitos de movimento em objetos espaciais.

Fica evidente, através dos estudos citados nessa seção, o avanço significativo das ferramentas de geoprocessamento, especialmente no contexto das aplicações Web GIS em 3D. *Frameworks* de código aberto, HTML5 e a própria WebGL foram, e ainda são, essenciais para a evolução de ferramentas GIS possibilitando a criação de ferramentas ainda mais sofisticadas, ampliando as possibilidades de representação e análise de dados geoespaciais e, principalmente, tornando essas ferramentas acessíveis para todos.

3. SOLUÇÃO PROPOSTA

A solução proposta representa uma significativa evolução do sistema base NodeGIS⁵, não apenas adicionando novas

⁵ <https://github.com/mateusqc/node-gis>

funcionalidades tridimensionais, mas também adaptando as funcionalidades preexistentes para funcionar na nova visualização de mapas em três dimensões. Além das funcionalidades já existentes, como por exemplo a plotagem de mapas vetoriais e as operações de *overlay*, os usuários têm a capacidade de fazer extrusão de polígonos, incluir camadas de hexágono e visualizar legendas dinâmicas com base nas camadas adicionadas e colunas escolhidas para determinação de elevação e/ou cor.

Essas melhorias são suportadas por uma arquitetura REST, dividida em *frontend* e *backend*, que será detalhada em seguida, proporcionando uma visão abrangente de como o sistema é estruturado e como as funcionalidades são implementadas.

3.1 Frontend

O frontend desempenha um papel fundamental na interação com o usuário e na apresentação dos dados geoespaciais de forma visualmente atraente e interativa. Para gerenciar o estado da aplicação de forma eficiente, foi utilizado o padrão Flux⁶ em conjunto com a biblioteca MobX⁷. O padrão Flux estabelece uma arquitetura unidirecional de fluxo de dados, garantindo uma atualização consistente dos componentes da aplicação. Os componentes do padrão Flux são: *Action*, *Store*, *Dispatcher* e *View*.

Combinado com o MobX, que é utilizado para simplificar a implementação do padrão Flux, permite-se substituir um *Dispatcher*, introduzindo um modelo de gerenciamento de estado mais reativo. Com o MobX os *stores* tornam-se observáveis, o que significa que as *views* são automaticamente atualizadas sempre que o estado do *store* muda. Isso elimina também a necessidade de enviar *actions* explicitamente e simplifica a lógica de atualização do estado da aplicação.

Além disso, o sistema é construído utilizando o *framework* React⁸, que oferece uma maneira eficiente de criar interfaces interativas e escaláveis, permitindo a criação de componentes reutilizáveis e a construção de uma interface modular, como também facilitando a manutenção do código e a implementação de novas funcionalidades.

O principal componente do *frontend* é o mapa, que no NodeGIS3D é renderizado com o uso da biblioteca deck.gl, que é uma ferramenta poderosa para visualização de dados geoespaciais, incluindo visualizações em três dimensões, permitindo a criação de mapas interativos e altamente personalizáveis. Ele é capaz de interpretar os dados em formato GeoJSON, simplificando o processo de integração e visualização dos dados no mapa.

Todos os dados utilizados no mapa são armazenados em seus respectivos *stores*, sendo obtidos por meio de requisições HTTP ao *backend*, que são realizadas utilizando a biblioteca Axios⁹, e então os dados geográficos são convertidos no formato GeoJSON para serem interpretados pelo deck.gl. Quando o usuário deseja adicionar uma nova camada vetorial ao mapa, uma requisição é feita ao *backend* para obter as informações das tabelas do banco de dados que possuem colunas do tipo geometria. As

⁶ <https://facebookarchive.github.io/flux/>

⁷ <https://mobx.js.org>

⁸ <https://pt-br.react.dev/>

⁹ <https://axios-http.com/>

configurações adicionais, como os dados exibidos no *tooltip* e a estilização da camada, são realizadas no frontend.

O frontend, além de ser responsável pela renderização do mapa, também é responsável pela implementação das funcionalidades de extrusão de polígonos e adição de camada hexagonal. A extrusão de polígonos proporciona uma representação tridimensional dos dados geoespaciais, adicionando uma dimensão vertical ao mapa e permitindo visualizar a magnitude de um atributo em relação à altura dos polígonos. Por sua vez, a adição de camada hexagonal oferece uma abordagem alternativa para visualizar e analisar dados, possibilitando a personalização da elevação com base em algum atributo específico e a atribuição condicional de cores, podendo inclusive utilizar um atributo distinto. Ambas as funcionalidades são configuradas no *frontend*, sendo então processadas, conforme as configurações estabelecidas pelo usuário e os dados correspondentes são interpretados e passados para as camadas do componente do *deck.gl*, para que possam ser renderizadas.

O NodeGIS3D possui a funcionalidade de poder manter salvo o estado atual do mapa, persistindo assim as camadas e seus respectivos metadados associados. Isso garante que, ao acessar novamente o NodeGIS3D, as camadas sejam exibidas conforme fora configurado anteriormente. A interface do usuário do NodeGIS3D é construída com o auxílio da biblioteca de componentes React chamada Ant Design¹⁰, que oferece uma ampla variedade de componentes que facilitam o desenvolvimento da interface.

Cada camada no mapa é representada por um objeto JavaScript¹¹ armazenado em uma estrutura do *store* do mapa, permitindo um acesso fácil e eficiente aos dados e metadados associados.

A Figura 1 mostra o formato que um objeto que representa uma camada segue, tendo a seguinte estrutura e atribuições:

- *name*: nome da camada;
- *key*: tabela do banco de dados que a camada representa;
- *type*: tipo de dado da coluna geométrica utilizada na camada;
- *geometryColumn*: coluna de geometria do banco de dados;
- *styles*: objeto que contém a definição de estilo estático para a camada, sendo utilizado para definição de cores dos polígonos (caso o atributo *styleType* seja igual a *static*);
- *styleType*: atributo que define qual o tipo de estilização dos polígonos da camada, se é estático (*static*) ou temático (*choropleth*);
- *choroplethStyleDefinition*: objeto que contém as definições de estilo para mapas temáticos, sendo utilizado para estilizar os polígonos (aplicável se o *styleType* estiver igual a *choropleth*);
- *displayColumns*: lista de objetos que definem as colunas que terão os seus valores exibidos no tooltip das feições da camada no mapa (no caso dos polígonos);
- *elevationScale*: multiplicador de elevação (utilizado para a extrusão dos polígonos caso a mesma esteja ativa);

- *extrudePolygon*: atributo que representa se a extrusão do polígono está ativa (*true*) ou não (*false*);
- *extrusionColumn*: coluna do banco que terá o seu valor utilizado para definir a elevação da extrusão do polígono (aplicável caso a extrusão esteja ativa);
- *hexagon*: atributo que representa se a camada hexagonal está ativa (*true*) ou não (*false*);
- *elevationColumn*: coluna do banco que terá o seu valor utilizado para definir a elevação do hexágono (aplicável caso a camada hexagonal esteja ativa);
- *elevationScaleHexagon*: multiplicador de elevação (utilizado para a elevação do hexágono caso a camada hexagonal esteja ativa);
- *radiusHexagon*: raio do hexágono em metros, utilizado na camada hexagonal caso a mesma esteja ativa;
- *choroplethStyleDefinitionHexagon*: objeto que contém as definições de estilo para a camada hexagonal (aplicável caso a camada hexagonal esteja ativa);
- *displayColumnsHexagon*: lista de objetos que definem as colunas que terão os seus valores exibidos no tooltip das “torres” da camada hexagonal;
- *data*: GeoJSON retornado pela consulta ao banco de dados e recebido pelo backend (por meio de requisição HTTP).

```
1  {
2    "name": "torres",
3    "key": "torres",
4    "type": "polygon",
5    "geometryColumn": "geom",
6    "styles": {
7      "fillColor": "#3388ff",
8      "fillOpacity": 0.7,
9      "color": "#3388ff",
10     "radius": 3,
11     "opacity": 1
12   },
13   "styleType": "static",
14   "choroplethStyleDefinition": {
15     "colorFunction": null,
16     "opacity": false,
17     "radius": null,
18     "defaultColor": "#3388ff",
19     "values": []
20   },
21   "displayColumns": [
22     {
23       "columnName": "torres",
24       "label": "torres",
25       "key": "torres"
26     }
27   ],
28   "elevationScale": 1,
29   "extrusionColumn": false,
30   "extrusionColumn": "",
31   "hexagon": false,
32   "elevationColumn": "",
33   "elevationScaleHexagon": 1000,
34   "radiusHexagon": 1000,
35   "choroplethStyleDefinitionHexagon": {
36     "colorFunction": null,
37     "opacity": false,
38     "radius": null,
39     "defaultColor": "#3388ff",
40     "values": []
41   },
42   "displayColumnsHexagon": [
43     {
44       "columnName": "torres",
45       "label": "torres",
46       "key": "torres"
47     }
48   ],
49   "data": {}
50 }
```

Figura 1: Objeto JSON que representa uma camada vetorial

¹⁰ <https://ant.design/>

¹¹ <https://www.javascript.com/>

Tais dados são utilizados para renderizar os elementos no mapa, sendo construídos componentes React do deck.gl. O componente principal é o *DeckGL*, que estabelece o contexto global do mapa na aplicação, definindo o ambiente onde os elementos correspondentes das camadas adicionadas serão renderizados. As camadas são adicionadas diretamente na propriedade *layers* do *DeckGL*, utilizando a estrutura de dados armazenada no estado *layers3D*. Os dados são obtidos diretamente do *Store* do mapa, possuindo um formato como mencionado anteriormente. Para as camadas de polígonos, os dados são passados para o componente *GeoJsonLayer*, que renderiza as feições no mapa conforme as instruções fornecidas. Quando a camada hexagonal estiver ativa, os dados pertinentes a ela são passados para o componente *HexagonLayer*.

A funcionalidade de consultas espaciais, funcionalidade já existente no NodeGIS, foi mantida, sendo possível realizar de duas formas: por consulta SQL e por seleção de feições do mapa. Apesar da consulta ter sua estrutura construída no *frontend*, os dados são enviados para o backend, para que seja feita no banco de dados.

3.2 Backend

O backend é construído conforme é feito no NodeGIS: opera como um servidor web, respondendo a requisições HTTP seguindo o padrão de API REST. Foi desenvolvido utilizando Node.js¹² em conjunto com o *framework* express¹³, sua arquitetura é dividida em três camadas principais: rotas, *controllers* e *repositories*. As rotas direcionam as requisições para os *controllers* correspondentes, que por sua vez gerenciam essas solicitações e suas respostas, utilizando dados do *repository* correspondente após o devido processamento. Este último é responsável por armazenar a lógica de construção de consultas espaciais, bem como o gerenciamento dos dados, incluindo interações com os bancos de dados relacionais espaciais.

O NodeGIS3D continua também oferecendo suporte a uma variedade de SGBDs (Sistema de Gerenciamento de Banco de Dados) relacionais espaciais, sendo: PostgreSQL¹⁴, MySQL¹⁵, MariaDB¹⁶, SQLite¹⁷ e CockroachDB¹⁸, possuindo um cliente específico para cada tipo suportado. Isso simplifica o acesso e o tratamento de possíveis divergências nas implementações SQL. Além disso, possui uma instância interna do SQLite para gerenciar os dados da aplicação, como as camadas persistidas pelo usuário e informações de conexão com outros bancos de dados.

Para realizar uma consulta espacial, o frontend envia uma requisição HTTP ao *backend* contendo os parâmetros necessários. Com base nessas informações, o backend constrói subconsultas que obtêm os dados solicitados, aplicando as operações espaciais correspondentes. Essas consultas são então convertidas para o formato GeoJSON, facilitando sua manipulação e representação no mapa.

¹² <https://nodejs.org>

¹³ <https://expressjs.com/>

¹⁴ <https://www.postgresql.org/>

¹⁵ <https://www.mysql.com/>

¹⁶ <https://mariadb.org/>

¹⁷ <https://www.sqlite.org/index.html>

¹⁸ <https://www.cockroachlabs.com/>

3.3 Containerização

Para facilitar o processo de implantação da aplicação, foram adotadas estratégias de containerização utilizando o Docker¹⁹. Enquanto o container PostgreSQL com PostGIS do sistema base NodeGIS foi reaproveitado no novo sistema, contêineres novos foram construídos para o *backend* e *frontend*.

No *frontend*, o build da aplicação React ocorre a partir da imagem base NodeJS com Alpine Linux²⁰. O Webpack²¹ e o Babel²² foram utilizados para o processo de *build* dos módulos JavaScript, resultando em arquivos JavaScript, CSS e HTML que são então movidos para uma imagem NGINX²³ que também está executando em Alpine Linux. O NGINX, por sua vez, serve esses arquivos como resposta às requisições HTTP.

Quanto ao *backend*, a imagem base NodeJS com Alpine Linux também é utilizada. Como se trata de uma aplicação NodeJS, não há necessidade de utilizar imagens Docker adicionais para sua construção. O conteúdo do diretório do *backend* no repositório é copiado diretamente para a imagem Docker, onde é executado pelo ambiente NodeJS.

Visando facilitar o processo de construção das imagens Docker, foi criado um arquivo *Makefile* contendo todos os comandos necessários para a criação de todas as imagens. Basta executar o comando “*make build*” na raiz do repositório para que seja iniciado o processo. Além disso, imagens da aplicação foram disponibilizadas publicamente no DockerHub²⁴, eliminando a necessidade de construí-las do zero.

Dois *scripts* Shell foram escritos e disponibilizados para facilitar o *deployment* do NodeGIS3D em ambientes com sistema operacional baseado em Unix: um para execução em ambiente local (*run-application-local.sh*) e outro para ambiente de produção com acesso público (*run-application-prod.sh*). Tais *scripts* já estão pré-configurados com parâmetros e variáveis de ambiente para cada contêiner, mas podem ser facilmente ajustados conforme necessário.

Além dos dois *scripts* Shell, foi disponibilizado mais um *script* PowerShell²⁵, este para execução em ambiente com sistema operacional Windows²⁶. Tal *script* (*run-application-local.ps1*) serve para a execução em ambiente local.

O terceiro contêiner, o PostgreSQL com PostGIS, que contém uma base de dados com tabelas de dados espaciais, foi reaproveitado do sistema base. Esse banco de dados é automaticamente configurado como padrão pela aplicação, facilitando sua utilização assim que a aplicação é executada.

4. CRIAÇÃO DE WEB GIS ATRAVÉS DO NODEGIS3D

O principal objetivo do NodeGIS3D é oferecer um sistema que una a facilidade de uso do seu antecessor, o NodeGIS, com

¹⁹ <https://www.docker.com/>

²⁰ <https://www.alpinelinux.org/>

²¹ <https://webpack.js.org/>

²² <https://babeljs.io/>

²³ <https://www.nginx.com/>

²⁴ <https://hub.docker.com/>

²⁵ <https://www.microsoft.com/PowerShell>

²⁶ <https://www.microsoft.com/pt-br/windows/>

funcionalidades que exploram o ambiente tridimensional. Além disso, proporciona um ambiente em que os usuários podem realizar operações de aproximação, distanciamento, rotação e inclinação da visualização. A containerização desempenha um papel crucial nesse contexto, simplificando a integração entre o *frontend*, *backend* e banco de dados. Para garantir uma experiência suave ao usuário, é necessário apenas ter o Docker pré-instalado.

4.1 Inicialização

Para facilitar a execução do NodeGIS3D, foram disponibilizados *scripts* na pasta raiz do repositório, permitindo a inicialização do sistema de forma rápida e simplificada, como foi explicitado na seção 3.3. Assim como foi disponibilizado no sistema base, o NodeGIS3D disponibiliza os dois *scripts* de inicialização para sistemas operacionais baseados em Unix, sendo o “run-application-local.sh” para ambiente local e “run-application-prod.sh” para ambiente de produção. Porém, foi adicionado também um *script* para inicialização em sistema operacional Windows, o “run-application-local.ps1”, para ambiente local.

Em sistemas operacionais baseados em Unix, o *script* pode ser executado com o comando “./run-application-local.sh”, por exemplo. Já no sistema operacional Windows, deve-se abrir o PowerShell e executar o comando “.\run-application-local.ps1”, porém antes de executar deve certificar que a *engine* Docker está em execução e que tenha a devida permissão para execução de *scripts*.

Os *scripts* de inicialização executam o comando “docker run” para cada contêiner (banco de dados, *frontend* e *backend*), comando este responsável por criar e executar os contêineres Docker necessários. Caso as imagens Docker não estejam presentes na máquina local, será feito o download das imagens que foram disponibilizadas publicamente no DockerHub automaticamente. Após o início dos contêineres, é exibido no terminal o código hash de cada contêiner, como também a URL de acesso ao NodeGIS3D.

Além dos *scripts* para inicialização da aplicação, foram disponibilizados dois *scripts* para a interrupção da execução do NodeGIS3D, sendo “stop-application.sh” para sistemas operacionais baseados em Unix e “stop-application.ps1” para Windows, a execução dos mesmos é de forma similar aos *scripts* de inicialização.

Para o ambiente de produção, que irá executar o *script* “run-application-prod.sh”, é importante notar que o ambiente deverá possuir IP público, além das portas 8080 e 8081 desbloqueadas.

É importante ressaltar, entretanto, que os parâmetros nos *scripts* podem ser alterados conforme necessário, como a troca de portas ou banco de dados, para se adequar ao ambiente específico de implantação.

Na Figura 2, é mostrada a disposição do sistema assim que é inicializado.



Figura 2: Sistema ao inicializar

4.2 Adicionando Camadas Vetoriais

Após a inicialização do NodeGIS3D, os usuários podem acessar as funcionalidades do sistema através do menu superior. Para adicionar as camadas vetoriais, os usuários deverão navegar até a opção “Add” e, em seguida, clicar na opção “Vector Layer” no submenu correspondente. Isso abrirá um modal onde os usuários podem escolher a tabela a ser utilizada, definir o nome que será dado para a camada a ser adicionada e selecionar a coluna geométrica. Além disso, como pode ser visto na Figura 3, estão disponíveis abas para personalizar a representação tridimensional das camadas, especificar o conteúdo do tooltip exibido nas feições e ajustar o estilo visual, incluindo cor de feições e espessura de linha. Essa abordagem oferece aos usuários um controle detalhado sobre a apresentação e interatividade dos dados geoespaciais no sistema.

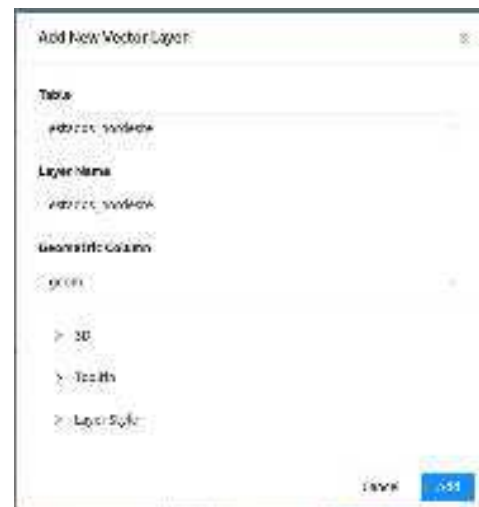


Figura 3: Modal da opção de adicionar camadas vetoriais

Ao clicar em “Add” no modal exibido, sem modificar nenhuma das opções fornecidas anteriormente, a camada é adicionada ao

mapa. As feições correspondentes são então exibidas, utilizando as cores e definições padrão do sistema, conforme pode ser visto na Figura 4. Dessa forma, o processo de adição de camadas é bem simples e o usuário consegue visualizar rapidamente as feições, sem a necessidade de fazer muitas configurações.



Figura 4: Camada de estados do nordeste do Brasil adicionada sem nenhuma configuração adicional

Além disso, o mapa atual pode ser salvo para que seja exibido automaticamente ao abrir o NodeGIS3D. Para tal, basta clicar no menu superior na opção “Save” e depois na opção “Map Current State”.

4.3 Utilização de funcionalidades 3D

No NodeGIS3D, é possível explorar funcionalidades tridimensionais para uma visualização mais imersiva dos dados geoespaciais. Através do mesmo modal de adição de camadas vetoriais, o que foi mostrado na Figura 3, na aba “3D”, são exibidas duas opções que são mutuamente exclusivas: “Extrusion” e “Hexagon”, como pode ser visto na Figura 5.



Figura 5: Modal de adição de camadas vetoriais com a opção “3D” expandida

Ao expandir a aba “Extrusion”, são mostradas as opções para o usuário configurar a extrusão de polígonos, como pode ser visto na Figura 6. Ao escolher ativar a opção “Polygon Extrusion”, o usuário pode definir a coluna que será utilizada para determinar a altura dos polígonos no mapa, como também é possível configurar o rótulo a ser exibido na legenda e ajustar a escala de elevação, que funciona como um multiplicador para ser aplicado ao valor da coluna escolhida para determinar a altura dos polígonos.

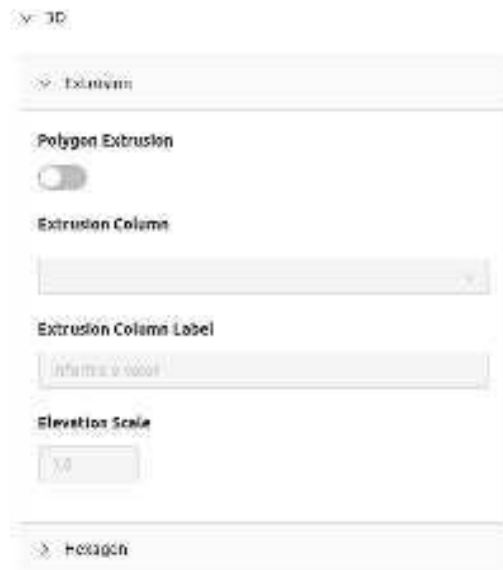


Figura 6: Opção “Extrusion” expandida

Na Figura 7, é mostrada uma camada com a extrusão ativada com elevação definida a partir do atributo de área, como pode ser visto, a legenda corresponde ao que foi configurado. Além disso, foi adicionada uma configuração de cor na camada, o que fez com que os polígonos tivessem cores distintas a depender do atributo perímetro.

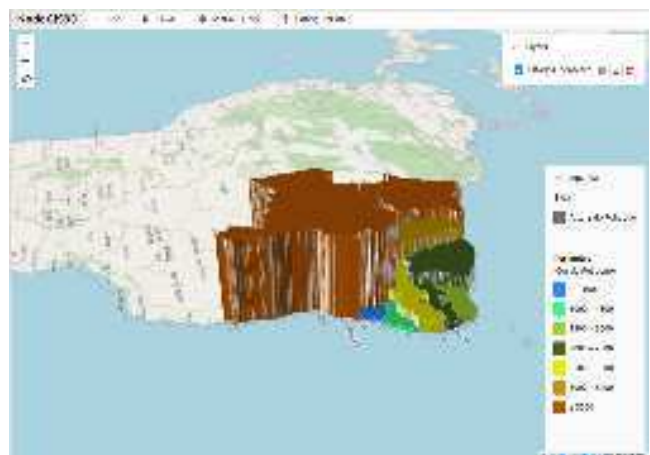


Figura 7: Camada adicionada com extrusão ativa

Já ao expandir a opção “Hexagon”, são mostradas as opções para o usuário configurar a adição de camada de hexágonos, como pode ser visto na Figura 8. Ao escolher ativar a opção “Hexagon Layer”, o usuário pode configurar diversos aspectos da camada hexagonal. Isso inclui a escolha da coluna que determinará a elevação dos hexágonos, o rótulo na legenda e a escala de elevação. Além disso, é possível definir o raio dos hexágonos, configurar o tooltip a ser exibido quando o mouse passar sobre os hexágonos e ajustar as cores. O botão “Set Color Variable” permite ao usuário personalizar as cores dos hexágonos, inclusive podendo condicioná-las aos valores de um atributo a ser escolhido da tabela para uma representação mais informativa dos dados.

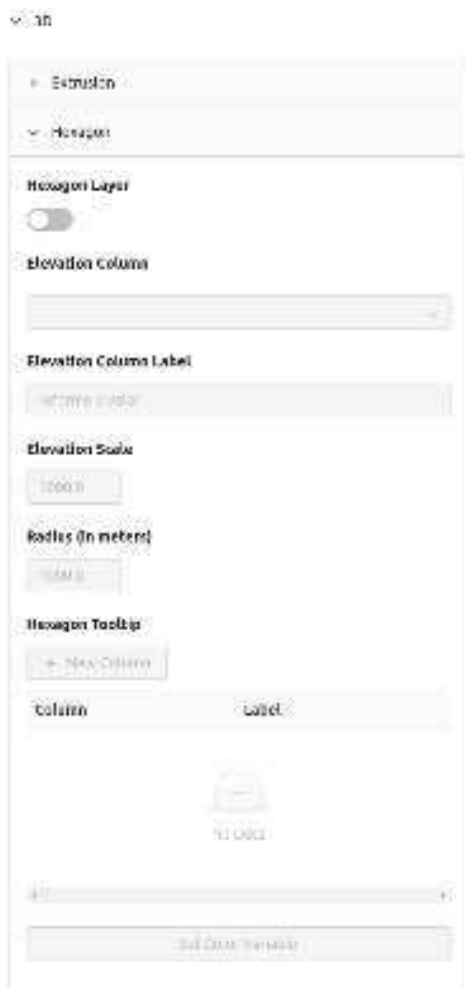


Figura 8: Opção “Hexagon” expandida

Na Figura 9, é mostrada uma camada com hexágono ativado, foi escolhido o atributo de população estimada para determinar a elevação do hexágono e foi determinada a cor dos hexágonos através de intervalos de valor do atributo densidade demográfica. Além disso, a cor do polígono foi configurada com base no atributo de área.

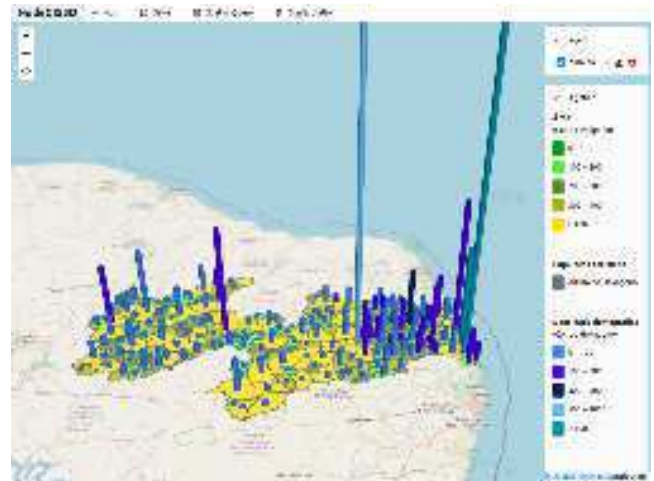


Figura 9: Camada adicionada com hexágonos

Também é possível configurar essas opções através do modal aberto ao clicar na opção de editar alguma camada já adicionada, acessível no menu posicionado na parte superior do lado direito, com o ícone de lápis ao lado do nome da camada.

5. AVALIAÇÃO DA FERRAMENTA

Para avaliar a usabilidade do NodeGIS3D, foi conduzida uma pesquisa utilizando um questionário baseado no PSSUQ (*Post-Study System Questionnaire*), versão 3, na qual participaram 17 respondentes.

Para testar o sistema, os participantes receberam a descrição de tarefas simples, incluindo a execução do sistema, adição de camada vetorial, ativação da extrusão de polígonos e adição de camada de hexágonos. Foi fornecido um manual detalhado²⁷ disponibilizado através do Notion, contendo passo a passo e pequenos vídeos demonstrativos.

Após a conclusão dos cenários, os participantes foram convidados a explorar livremente o sistema sem supervisão adicional. Em seguida, foi apresentado o questionário, composto por uma pergunta sobre o conhecimento prévio de SIG e cinco afirmações relacionadas à usabilidade do NodeGIS3D, sendo elas:

- Fui capaz de completar as tarefas e cenários de forma satisfatória;
- Considero as funcionalidades do sistema intuitivas;
- A disposição das funcionalidades e informações nas telas foi satisfatória;
- Recomendaria o sistema a outras pessoas;
- No geral, estou satisfeito com o sistema.

Todas as afirmações obedeciam uma escala Likert, com cinco valores, variando de “Discordo totalmente” (1) a “Concordo totalmente” (5).

O resultado da pergunta sobre o conhecimento prévio de SIG pode ser visto na Figura 10. Mostrando que 47,1% dos participantes afirmaram não possuir conhecimento prévio, enquanto 52,9%

²⁷ <https://bit.ly/4doHLhP>

declararam ter algum nível de familiaridade com SIG. Essa distribuição equilibrada reflete uma amostra representativa de usuários com diferentes níveis de experiência em SIG, permitindo uma avaliação abrangente da usabilidade do NodeGIS3D em diferentes contextos de uso.



Figura 10: Resultado da pergunta “Possuo conhecimento prévio de SIG”

Na Figura 11, podem ser encontrados os demais resultados do questionário. A grande maioria dos participantes atribuiu pontuação máxima em todas as afirmações, demonstrando uma avaliação positiva da usabilidade da ferramenta. De forma que todos os participantes atribuíram nota igual ou superior a 4 para todas as afirmações, e a pontuação máxima (5) foi maioria nas avaliações de cada afirmação.

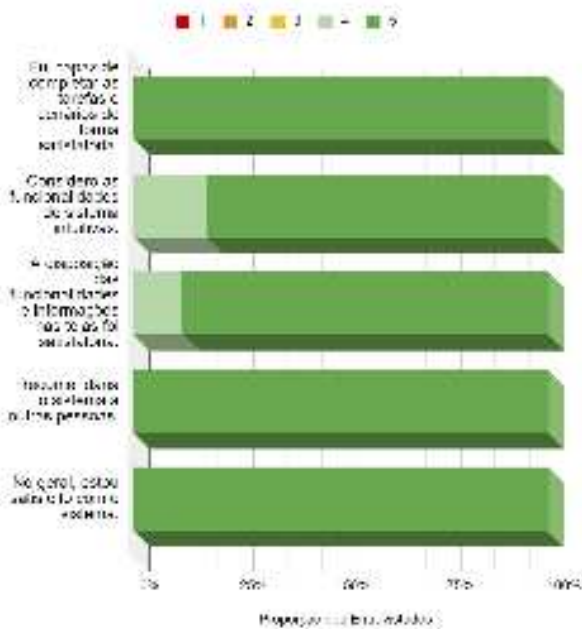


Figura 11: Resultado geral do questionário de usabilidade

A média geral da avaliação obtida com o questionário foi de aproximadamente 4,94, indicando uma alta satisfação dos usuários com o NodeGIS3D, já que a nota máxima seria 5. Tal pontuação reflete a percepção positiva dos participantes em relação ao sistema.

6. CONSIDERAÇÕES FINAIS

O desenvolvimento do NodeGIS3D representa um avanço significativo em relação ao sistema NodeGIS, oferecendo aos usuários uma ferramenta mais robusta para visualização e análise de dados geoespaciais em três dimensões. Ao longo deste artigo foram exploradas as principais funcionalidades e características do NodeGIS3D.

A mudança para um ambiente que trabalha com as três dimensões possibilita uma ampla gama de novas funcionalidades e oportunidades para a ferramenta. Portanto o NodeGIS3D abre portas para uma análise mais aprofundada e uma compreensão mais completa dos dados geoespaciais.

A adição das funcionalidades de extrusão de polígonos e de camada hexagonal oferece novas perspectivas e *insights* sobre os dados, enriquecendo ainda mais a experiência do usuário.

Durante o desenvolvimento da ferramenta, foi enfrentado um desafio ao integrar a biblioteca *deck.gl* em substituição à tecnologia utilizada anteriormente. Essa mudança não apenas exigiu a substituição da tecnologia, mas também a adaptação das camadas existentes para funcionarem de maneira eficaz com *deck.gl*. Além disso, a adição de camadas tridimensionais representou outro desafio, pois era uma funcionalidade nova que demandava estudo e compreensão detalhada. No entanto, todos esses desafios foram superados por meio de uma análise cuidadosa da documentação do *deck.gl* e do estudo do código existente para compreender seu funcionamento.

O desenvolvimento da ferramenta foi realizado com entregas semanais, como acontece em *sprints* de metodologias ágeis. A cada semana, eram realizadas entregas incrementais e eram apresentadas as funcionalidades desenvolvidas até então, proporcionando transparência e *feedback* contínuo. Essa abordagem permitiu manter o sistema sempre funcional e utilizável, além de facilitar a identificação de possíveis ajustes e priorização de novos recursos para as próximas iterações.

Apesar dos avanços alcançados, ainda há espaço para melhorias e aprimoramentos contínuos. A integração de mais recursos de análise avançados, nuvem de pontos, adição de camadas de contorno e de ícones são áreas que podem ser exploradas em trabalhos futuros.

AGRADECIMENTOS

Gostaria de expressar meus agradecimentos a todas as pessoas que contribuíram de alguma forma para a realização deste trabalho. Em primeiro lugar, agradeço a Deus pela orientação e pelas oportunidades concedidas ao longo dessa jornada. Ao meu orientador, professor Cláudio, pela sua dedicação, orientação e valiosas contribuições que foram fundamentais para o desenvolvimento deste projeto. Agradeço também a todos os professores do curso de Ciência da Computação da Universidade

Federal de Campina Grande (UFCG), cujo conhecimento e dedicação foram fundamentais para a minha formação acadêmica.

Agradeço ainda aos meus colegas do Laboratório de Sistemas da Informação (LSI), pela troca de experiências, pelo apoio mútuo e pelo ambiente que tornou possível o meu crescimento profissional e pessoal. À minha família, pelo amor incondicional, pelo incentivo constante e pelo suporte emocional em todos os momentos. Aos meus amigos, pela compreensão, pelo apoio e pela motivação durante toda a jornada acadêmica.

REFERÊNCIAS

- [1] CUNHA, Mateus Queiroz. NodeGIS: simplificando o desenvolvimento de aplicações web de geoprocessamento. 2021. 14f. Trabalho de Conclusão de Curso (Artigo), Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande - Paraíba - Brasil, 2021. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/25017>
- [2] QIU, G.; CHEN, J. Web-based 3D map visualization using WebGL. 1 maio 2018.
- [3] METE, M. O.; GULER, D.; YOMRALIOGLU, T. DEVELOPMENT OF 3D WEB GIS APPLICATION WITH OPEN SOURCE LIBRARY. Selcuk University Journal of Engineering ,Science and Technology, v. 6, n. Özel (Special), p. 818–824, 31 dez. 2018.
- [4] MIAO, R.; SONG, J.; ZHU, Y. 3D geographic scenes visualization based on WebGL. 1 ago. 2017.
- [5] GESQUIÈRE, G.; MANIN, A. 3D Visualization of Urban Data Based on CityGML with WebGL. International Journal of 3-D Information Modeling, v. 1, n. 3, p. 1–15, jul. 2012.
- [6] CHRISTEN, M.; NEBIKER, S.; LOESCH, B. Web-Based Large-Scale 3D-Geovisualisation Using WebGL. International Journal of 3-D Information Modeling, v. 1, n. 3, p. 16–25, jul. 2012.
- [7] EVANGELIDIS, K. et al. 3D geospatial visualizations: Animation and motion effects on spatial objects. Computers & Geosciences, v. 111, p. 200–212, fev. 2018.