



WebMapa: Desenvolvimento de Mapas Web por um Complemento QGIS

WebMap: Web Map Development by a QGIS Complement

Fabrcio Rosa Amorim¹ e Marcio Augusto Reolon Schmidt²

¹ Universidade Federal do Paran, Programa de Ps-Graduao em Cincias Geodsicas, Curitiba, Brasil.
fabrcioamorimeac@hotmail.com

ORCID: <https://orcid.org/0000-0002-6670-2131>

² Universidade Federal do Paran, Programa de Ps-Graduao em Cincias Geodsicas, Curitiba, Brasil.
marcio.schmidt@gmail.com

ORCID: <https://orcid.org/0000-0003-2716-2360>

Recebido: 10.2020 | Aceito: 04.2021

Resumo: Este trabalho descreve o processo de construo e validao de um (a) *plugin* / ferramenta em Sistema de Informaoes Geogrficas (SIG), utilizado para desenvolvimento de mapas *Web* / *online*. O *plugin*, denominado WebMapa, ainda em desenvolvimento, permite ao usurio adicionar elementos cartogrficos s pginas da *Web*, tornando-os interativos e funcionais. O *plugin* WebMapa foi construdo no *software* QGIS, utilizando a linguagem de programao *python*. O sistema foi desenvolvido como um mecanismo de busca que permite aos usurios escolherem informaoes e elementos de mapas, bem como simboliz-los. As escolhas possveis so os seguintes elementos de mapas: base cartogrfica (ex.: OpenStreetMap, Google Maps), bssola, escala, localizao do usurio no mapa, edio em tela, cculo de rota entre pontos de partida e chegada. O resultado do processamento do *plugin* WebMapa foi construdo como um *Web site* disponvel no formato de sada *.html* (HTML - *Hypertext Markup Language* / Linguagem de marcao de hipertexto). Desta forma, o sistema se torna multiplataforma, uma vez que pode ser acessado como um web site em qualquer tipo de navegador Web independente do sistema operacional.

Palavras-chave: Sistemas de Informaoes Geogrficas (SIG). QGIS. Mapas Web.

Abstract: This work aims to describe the construction process and the validation of a *plugin* / tool in Geographic Information Systems (GIS), used for the development of Web maps. For this, the WebMapa *plugin*, still under development, allows adding cartographic elements to Web pages, making them interactive and adaptable. The GIS was built as a complement (WebMapa *plugin*) to the QGIS software, using a python programming language. The *plugin* was developed as a search engine that allows users to choose information and map elements, as well as symbolize them. The options made possible the choices of the following map elements: cartographic base (eg: OpenStreetMap, Google Maps); compass; scale; user's location on the map; screen editing; route calculation between starting and ending points. The result of the processing of the WebMapa *plugin* was built as a website available in the *.html* output format (HTML - Hypertext Markup Language). In this way, the system becomes multiplatform since it can be accessed as a Website in any type of Web browser, regardless of the operating system.

Keywords: Geographic Information Systems (GIS). QGIS. Web maps.

1 INTRODUO

No mais restritos s representaoes em papel, os mapas so atualmente produzidos, transmitidos e disponibilizados aos usurios em menor tempo. Alm disso, a Internet possibilita a visualizao, a manipulao e a disseminao de dados espaciais e mapas de modo mais interativo e rpido. Porm, ainda so necessrias pesquisas para tornar a Internet um meio efetivo de transmisso da informao espacial na forma de mapas (ICA, 2007). Na cartografia, geralmente, a interface  o prprio mapa, uma vez que o fenmeno que est sendo representado  o mesmo sobre o qual so realizadas a navegao e a interao (FALAT; DELAZARI, 2010). As novas formas de abordagens com mapas podem ser definidas como processos de Visualizao Cartogrfica ou Geovisualizao. Para tanto, a Geovisualizao “ uma integrao da visualizao cientfica, cartografia exploratria, anlise de imagens, visualizao da informao, anlise exploratria de dados e Sistemas de

Informação Geográfica (SIG) para produzir teorias, métodos e ferramentas para a exploração visual, análise, síntese e apresentação de dados geoespaciais” (MACEACHREN; KRAAK, 2001).

Com o avanço da Internet, os sistemas de informações passaram a ser mais atrativos devido aos seus baixos custos de desenvolvimento, implantação e manutenção (AMORIM et al., 2008). Nos últimos anos, principalmente depois que o Google lançou o Google Maps (em fevereiro de 2005) e o Google Earth (em junho de 2005), há um interesse cada vez maior em empregar tais plataformas para a identificação, localização, orientação, navegação e mensuração, tanto sobre representações 2D quanto 3D (MENEGETTE, 2014). Para Kraak (2010), com a mudança para ambiente Web é necessário pesquisar o uso do mapa e as necessidades do novo usuário. Com o conhecimento destas necessidades, as ferramentas de acesso aos dados espaciais poderão ser adaptadas de forma a atendê-las. Nesse sentido, Kraak e Brown (2003) e Marisco et al. (2004) destacam que os produtos cartográficos na Web são influenciados por alguns fatores, sendo eles: os usuários; os provedores de dados; o ambiente de concepção dos produtos cartográficos; o conteúdo dos produtos cartográficos; a tradução gráfica; a interface com o usuário; e o armazenamento dos dados geoespaciais.

Atualmente, os SIG veem acompanhando a evolução tecnológica e incorporando novos paradigmas que auxiliam o processo de integração entre sistemas. Um exemplo disso é a migração dos dados armazenados em arquivos para banco de dados espaciais (TSOU, 2003). Neste sentido, os Serviços Web são uma proposta da comunidade científica para integrar sistemas geográficos distribuídos (ALAMEH, 2002; MELO JR; CANDEIAS, 2005, 2010). Os sistemas cartográficos baseados na Internet, em sua maioria, são compostos por Web *sites* interativos e dinâmicos em conjunto com um banco de dados localizado no servidor remoto de arquivos. Em geral, o usuário faz a requisição de informações através do Web *site*, este se comunica com o banco de dados e retorna os valores em uma página de resposta. A interação poderá ser ampla ou restrita, dependendo do tipo de informação a ser disponibilizada ao usuário e das suas funcionalidades interativas. Os símbolos gráficos em um produto cartográfico interativo podem se tornar objetos ativos, clicáveis e recuperáveis de um servidor de banco de dados. As legendas podem ser apresentadas em janelas secundárias “*pop-up*”, as cores podem ser alteradas, objetos em movimento podem ser disponibilizados, diferentes níveis de informação em diferentes escalas podem ser apresentados por meio de funções Zoom (*in-out*) (ampliações e reduções).

Nos campos da cartografia on-line e interativa, a programação costuma ser uma parte importante da tarefa de criar um mapa (LEDERMANN; GARTNER, 2015). Visto que os cartógrafos não se limitam a um papel de consultoria em contextos tecnológicos e não deixam a implementação para os programadores, cientistas da computação, etc., as bases tecnológicas para a criação de mapas por meio de código são essenciais (LEDERMANN; GARTNER, 2015). Esses mapas raramente são programados "do zero" usando apenas funções de baixo nível fornecidas pelo ambiente de execução (por exemplo, sistema operacional ou navegador da Web); uma API (*Application Programming Interface* - Interface de Programação de Aplicativos) pode fornecer partes reutilizáveis de código, expostas como um conjunto de abstrações e métodos de alto nível, para o criador do mapa. Usando uma API apropriada, a configuração de processos tecnologicamente complexos pode ser reduzida a linhas únicas de código, e essa alavancagem pode ser usada para implementar mapas interativos com recursos ou conhecimento que não seriam suficientes para criar uma solução idêntica com código de baixo nível. Automatizar esse processo pode corroborar na dinamicidade dos processos cartográficos para a construção de mapas, sobretudo, os mapas *online*.

Segundo Cartwright (2008), com o advento da Internet, a cartografia pôde publicar mapas *online*, com rapidez e sem custos de distribuição ou impressão. No entanto, a maneira como os mapas são construídos para entrega na Web difere pouco da computação gráfica e de produtos de multimídia discretos. Agora, acelerado pelo acesso relativamente barato à Internet, a disponibilidade de computadores móveis pequenos e baratos e a disponibilidade de software social mudaram a maneira como os usuários acessam as informações pela Internet. Isso tem sido chamado de Web 2.0 (O'REILLY, 2005). Peterson (2015) comparou diferentes APIs de “mapas deslizantes” (ex.: API do Google Maps, Leaflet, Mapstraction), aplicando critérios quantitativos (velocidade de execução, comprimento do código, custo) a programas de exemplo. Embora a “funcionalidade cartográfica” seja listada como um critério de avaliação, nenhuma exploração sistemática da funcionalidade das diferentes APIs e como elas se relacionam com os conceitos cartográficos é fornecida. Peterson conclui com uma posição

pragmática, mencionando o custo e a viabilidade a longo prazo como possivelmente os critérios de avaliação mais importantes para contextos de produção - ambos os quais não estão relacionados com a funcionalidade fornecida pela API.

Em uma revisão de literatura, especialmente brasileira, não foram encontrados trabalhos que objetivassem descrever o processamento de mapas Web com o uso de plataformas abertas de SIG sem a necessidade de desenvolvimento de algoritmos de processamento. No entanto, podemos encontrar algumas ferramentas implementadas no QGIS responsáveis por gerar mapas Web, tais como o *qgis2web* e o *Qgis2threejs*. O objetivo desse trabalho foi, através das ferramentas de geotecnologia, otimizar as atividades de produção de mapas Web, as quais podem auxiliar no processo de navegação por mapas. A hipótese deste trabalho centra-se na assertiva de que é possível utilizar fontes abertas em projetos de mapas para Web interativos, com o intuito de visualizar e interagir com informações previamente estabelecidas, sem a necessidade de o usuário desenvolver linhas de código. Por meio da utilização de um SIG, denominado QGIS, o qual apresenta o código aberto, desenvolveu-se um *plugin* (“WebMapa”) para facilitar o projeto de um produto cartográfico interativo na Web. Para validar a hipótese, utilizou-se informações concernentes à uma área piloto, localizada na Universidade Federal do Paraná – UFPR, campus Jardim Botânico.

2 MATERIAIS E MÉTODO

2.1 Materiais

Utilizou-se duas bases de dados e três softwares livres com diferentes ferramentas, ícones e bibliotecas de linguagem de programação, mais especificamente:

- Base de Dados: OpenStreetMap; CampusMap;
- *Softwares*: QGIS (complementos: Processing Modeler; Plugin Reloader; Plugin Builder); Atom (bibliotecas *python*: PyQt4.QtCore; PyQt4.QtGui; Osgeo; Folium; Requests; Gdal; Processing; Psycopg2); Qt Designer (ícones: Labels; QDialogBox; QLineEdit; SpinBox; QFileDialog).

2.2 Método

Dos objetivos a serem alcançados na pesquisa destacam-se: Analisar a utilização das novas tecnologias disponíveis para Web em códigos padronizados, dentro do contexto da geovisualização. Investigar os diferentes formatos e modelos de dados que possam ser utilizados na construção de mapas interativos para Web. Desenvolver um sistema utilizando códigos padronizados e fontes abertas para projeto de mapas para Web. Desenvolver um protótipo de um mapa para Web que seja uma interface de comunicação com o usuário. Fornecer uma contribuição ao processo de ampliação e de popularização ao acesso às informações.

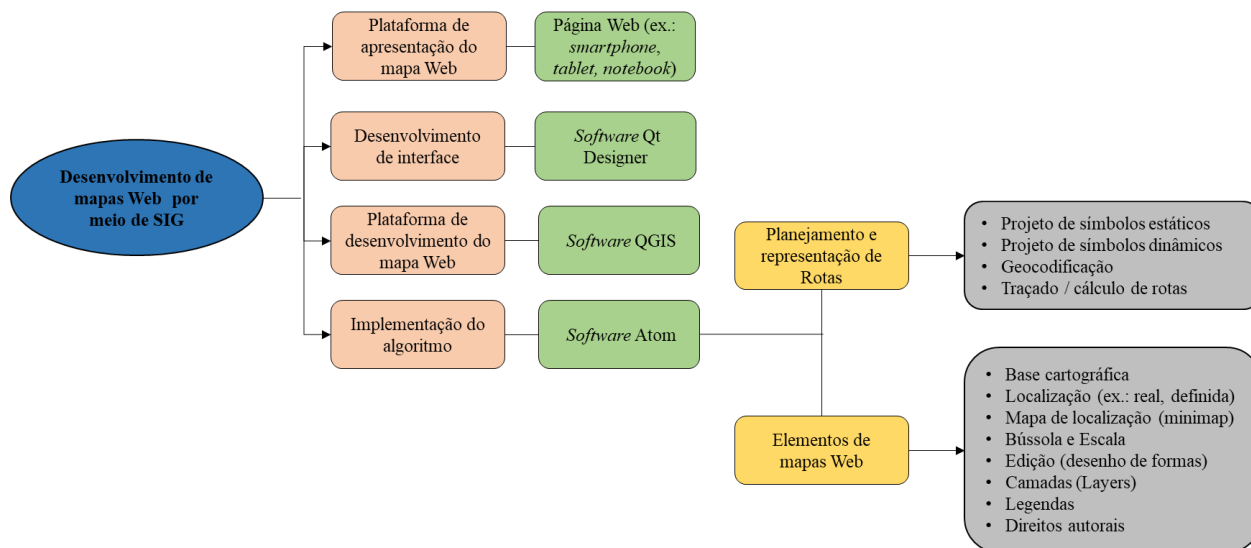
Van Elzaker (1999) propõe um número mínimo de requisitos para um programa eficientemente realizar a visualização de dados: funcionalidades multijanelas; variáveis visuais dinâmicas; sistema cartográfico especialista; comparações espaciais; generalização; metadados; modelagem espacial; interface gráfica interativa; e capacidade de edição via imagem. Stylos e Myers (2007) e Ledermann e Gartner, (2015) apontam a importância de um modelo mental representar um processo cartográfico. Para o modelo mental central da popular API do Google Maps - mostrar um mapa de uma área, adicionar marcadores e conteúdo ao mapa - pode-se argumentar que essa visualização não reflete o modelo mental de um cartógrafo, mas sim de um usuário do mapa (LEDERMANN; GARTNER, 2015).

A metodologia aqui proposta visa desenvolver uma ferramenta para visualização de um mapa interativo Web exibido em conjunto às camadas de informações. Para tanto, as principais funcionalidades do sistema são oriundas dos exemplos *python* fornecidos na página do Folium (2020a, 2020b). A Figura 1 apresenta as etapas para o desenvolvimento do *plugin*, divididas em duas plataformas de utilização (QGIS para abarcar o *plugin* e página Web para visualizar o mapa final) e duas etapas de desenvolvimento (interface, algoritmo) com os respectivos softwares utilizados. As principais funcionalidades empregadas na implementação do algoritmo são descritas em duas etapas, o planejamento de rotas e a utilização de elementos de mapas (ex.: legenda, escala). Para tanto, o *software* utilizado para abarcar o sistema desenvolvido foi o

QGIS, o qual é *opensource* (dados abertos).

O sistema (no geral, para desenvolvimento e visualização) foi implementado usando a linguagem de programação *python*, a qual se comunica com as linguagens JavaScript e HTML durante alguns processos. Utilizou-se o *software* Atom para implementar as linhas de código do algoritmo. A interface do sistema desenvolvido foi projetada no *software* QtDesigner, disponível automaticamente ao se realizar o *download* do QGIS. Para visualizar os resultados obtidos no QGIS após a utilização do sistema, o mapa interativo Web pode ser acessado como uma página da Internet, em algum servidor HTTP (ex.: *Google Chrome*).

Figura 1 – Ferramentas utilizadas para o desenvolvimento do *plugin* WebMapa.



Fonte: Os autores (2020).

A base cartográfica pode mostrar o contexto geográfico, tais como: estradas, cidades e pontos de referência. A base cartográfica depende de Biblioteca de mapeamento *LeafletJS* para fornecer os *TileLayers*. O Folium é baseado na extensa biblioteca JavaScript “*Leaflet*” e foi usada para gerar mapas personalizados com código que funde dados para o *TileLayer* selecionado. Dentre as funcionalidades de mapas permitidas pelo Folium, destacam-se: visualização de bases cartográficas contínuas do mundo; adicionar e agrupar feições, tais como objetos vetoriais *GeoJSON* e *shapefiles* estáticos e dinâmicos (para linhas); projeto de mapas coloridos (coroplético; corocromático e mapa de calor ou densidade de kernel, com ou sem controle de tempo); controles da escala e da extensão do mapa; adicionar símbolos pontuais personalizados, tais como marcos com ou sem cluster, podendo rotacioná-los; adicionar imagens georreferenciadas ou não (arquivos de formato *.jpg* e *.png*) e gráficos; adicionar pop-up às feições (janela de informações ao selecionar a feição); adicionar minimap (mapa de localização); a possibilidade de editar formas (ex.: ponto, linha, polígono) sobre o mapa e salvá-las na memória interna do dispositivo de mídia ao qual estiver utilizando (em formato de arquivo *.geojson*); e medição de áreas e comprimentos.

Alguns trabalhos que utilizam o Folium podem ser encontrados na literatura, dentre os quais: Chourasia et al. (2017) para visualização de Times Series EarthQuake Data; Samreth et al. (2018) para visualização de casos de alergias; Sarajcev e Matijasevic (2018) para análise de redes; Pappalardo et al. (2019) para visualização de dados de mobilidade urbana; e Jawa et al. (2020) para visualização de dados de crimes. O Folium foi utilizado no *plugin* desenvolvido para fornecer os elementos e ferramentas do mapa Web, dentre as diversas opções possíveis, utilizou-se: as opções de bases cartográficas (OpenStreetMap, Google Maps, Google Earth, Stamen Terrain, Stamen Toner, Stamen Watercolor, CartoDB Positron, CartoDB Dark_Matter); a localização do usuário em tempo real; a localização central do mapa (a partir de dados de entrada fornecidos pelo usuário e geocodificados); o mapa de localização (minimap), a bússola, a escala e os direitos autorais; edição em tela para desenho e exportação de formas; a adição e simbolização de camadas (*Layers*); e a adição de legendas, logo ou imagem sobre o mapa a partir de um arquivo em formato de imagem (ex.: *.png*, *.tif*).

Foram tomados alguns cuidados referentes à privacidade do usuário e aos direitos autorais da base

cartográfica. Os dados de localização de um usuário em algum dispositivo só podem ser acessados após explícito consentimento do mesmo. Para tanto, para utilização da latitude e da longitude em tempo real do usuário, uma janela solicitando a permissão do usuário é disponibilizada da tela. Os direitos autorais são uma forma essencial pela qual os designers e produtores de mapas podem proteger seus direitos (Cartwright, 2008). Nesse sentido, em acordo com a base cartográfica selecionada pelo usuário para utilização em seu mapa, os direitos autorais da imagem são descritos no canto inferior direito da tela.

Para utilizar as coordenadas pertencentes à centralização do mapa ou pertencentes ao início e final da rota, as mesmas são consultadas por meio do processo de geocodificação com base em um servidor remoto que utiliza a biblioteca *python* denominada Geopy. Para o traçado das rotas, os dados de vias foram obtidos através da base cartográfica vetorial do OpenStreetMap. O cálculo da rota foi realizado com a utilização do algoritmo nativo do QGIS denominado *shortestpathpointtopoint*, o qual determina o menor caminho entre dois pontos de coordenadas conhecidas sobre uma rede. A simbolização das camadas (formatos vetoriais de ponto, linha e polígono) a serem adicionadas ao mapa foi realizada utilizando as possibilidades do Folium. Além da utilização da biblioteca *python* Geopy e do algoritmo QGIS *shortestpathpointtopoint*, as inúmeras bibliotecas *python* e algoritmos do QGIS (ex.: GDAL, GRASS e SAGA) podem ser utilizados na criação de produtos cartográficos digitais, tal como o descrito nesse trabalho.

Para a camada de pontos, é possível o usuário determinar: o formato do ícone (pin ou apenas o símbolo); o pop-up contendo uma mensagem caso o usuário clique sobre o ponto; a cor do ícone; o formato do símbolo localizado dentro do ícone; a cor do símbolo; e o tamanho do símbolo. Para a camada de linhas, a simbolização consiste na escolha: da cor de preenchimento; da espessura da linha; da cor da borda; e da espessura da borda. Para a simbolização da camada de polígonos, é possível ao usuário determinar: a cor de preenchimento; a transparência do preenchimento; a cor da borda; e a espessura da borda. Para a simbolização da rota, é permitido ao usuário determinar o padrão do símbolo pontuais de início / partida e fim / chegada da mesma, para tal, utilizou-se as mesmas propriedades de simbolização das camadas de ponto. Ainda, para representar a linha que liga esses pontos, o usuário pode selecionar as mesmas características possíveis para adição de camadas de linha. Além desse estilo de simbolização (estática), também é possível a representação de rotas com símbolos pontuais dinâmicos sincronizadamente se locomovendo sobre a linha utilizando a função *AntPath* da biblioteca *Folium*. Para tal dinamicidade, o usuário pode escolher o formato e a cor do símbolo dinâmico, bem como escolher a velocidade e o espaçamento.

Após a escolha dos elementos cartográficos pelo usuário, o mapa final é gerado para ser visualizado como uma interface gráfica utilizando controles compatíveis com dispositivos móveis e *desktops*, e que estão nativamente disponíveis com / como arquivos HTML. Portanto, o mapa é gerado e visualizado usando um servidor HTTP personalizado. O cliente (navegador da Web) busca os dados servidos e exibe o resultado junto com a interface do usuário.

3 RESULTADOS

A interface do usuário (visualizada no QGIS) é intuitiva, sua função é definir os dados necessários como entrada para visualização do mapa Web. Essa interface contém um elevado número de funções ou opções, portanto, necessitou-se cuidados com o design da interface visando facilitar sua utilização pelos usuários. Portanto, a interface foi subdividida em três seções relacionadas com suas principais funcionalidades. A primeira seção contém as opções de selecionar a base cartográfica do mapa, o nível de zoom do mapa (de 1 à 20 – da escala global à escala local), o local para salvar o arquivo final em formato de arquivo *.html*, a localização inicial ou central do mapa com a devida simbolização desse ponto, e as opções para simbolização da rota (caso selecionada) (Figura 2). A segunda seção da interface permite ao usuário escolher as ferramentas do mapa (ex.: mapa de localização e edição de formas) (Figura 3). A terceira seção permite ao usuário escolher e simbolizar as camadas vetoriais para sobreposição ao mapa (formatos de ponto, linha e polígono).

Figura 2 – Primeira seção da interface, tela inicial (a) e tela inicial com as opções para simbolização da rota (b).

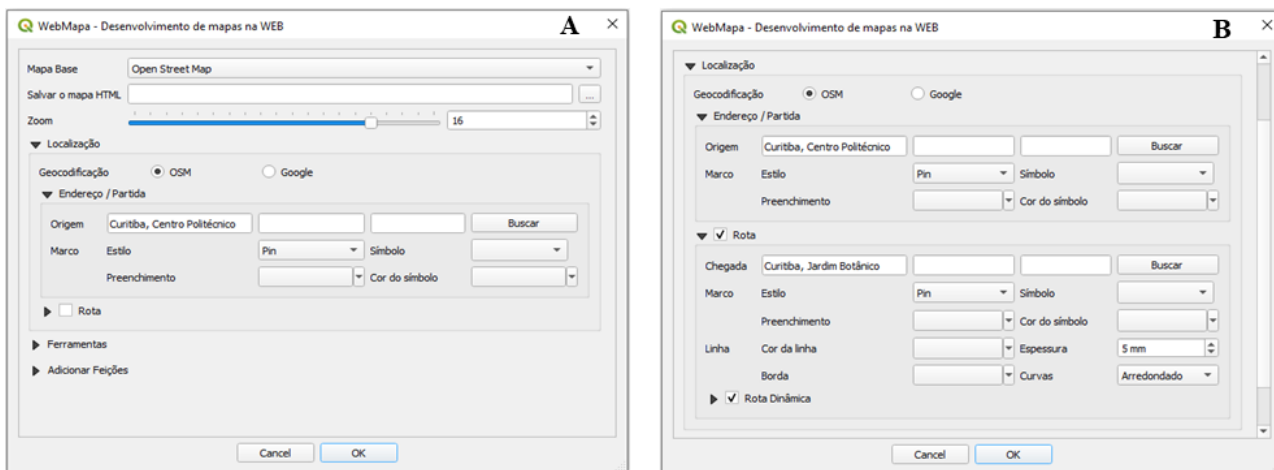
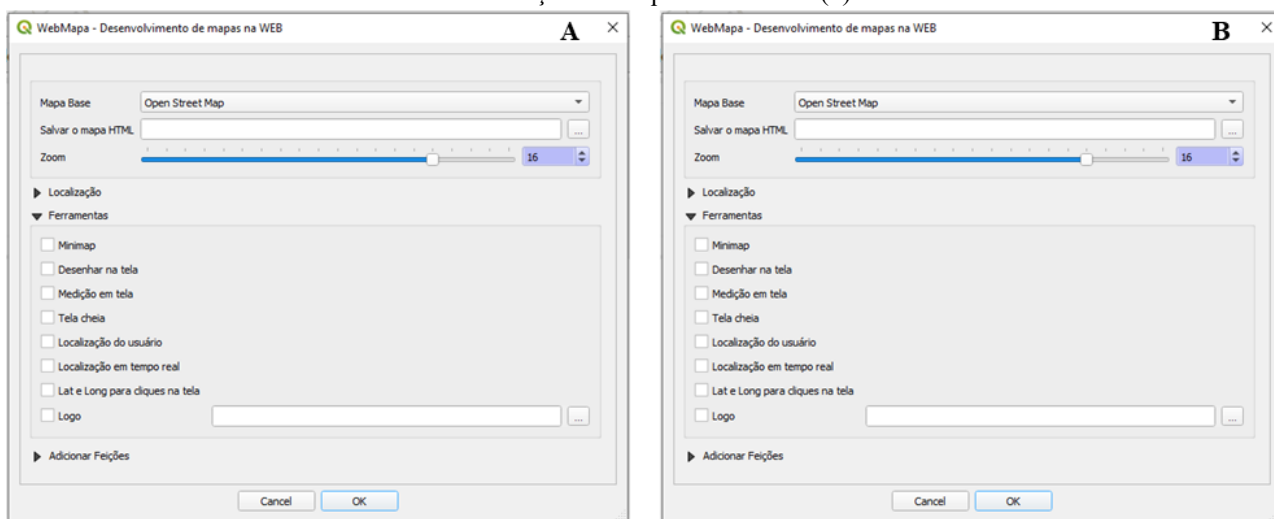


Figura 3 – Segunda seção da interface com as opções de ferramentas do mapa (a) e terceira seção com as opções para simbolização dos arquivos vetoriais (b).

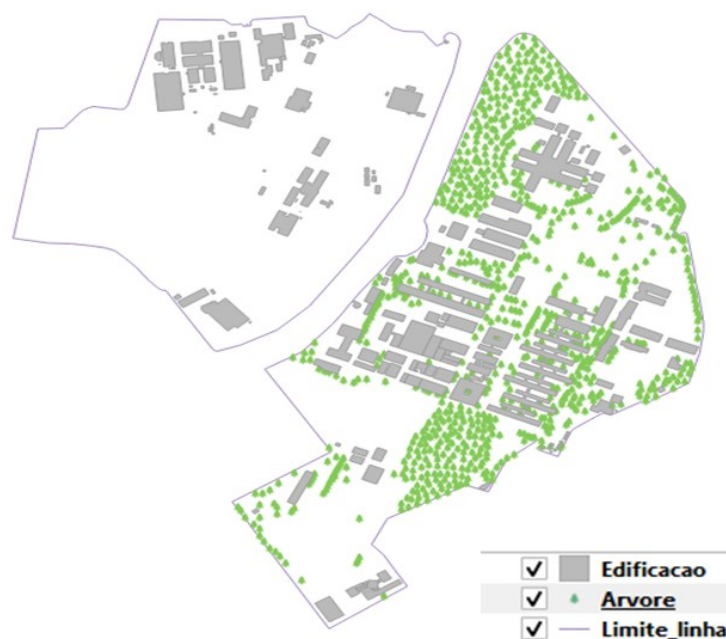


Fonte: Os autores (2021).

3.1 Validação dos Resultados

Para validar os resultados alcançados com o desenvolvimento do complemento QGIS denominado plugin *WebMapa*, uma área teste foi utilizada nos experimentos. A área escolhida para o ensaio metodológico corresponde à Universidade Federal do Paraná, campus Centro Politécnico, Curitiba-PR. Os dados utilizados para realizar os testes foram obtidos através da base cartográfica do CampusMap. Para tal, utilizou-se as camadas de árvores (ponto) e limites da UFPR (linha) edificações (polígono), como apresentado na Figura 4.

Figura 4 – Camadas vetoriais utilizadas para a validação dos resultados.



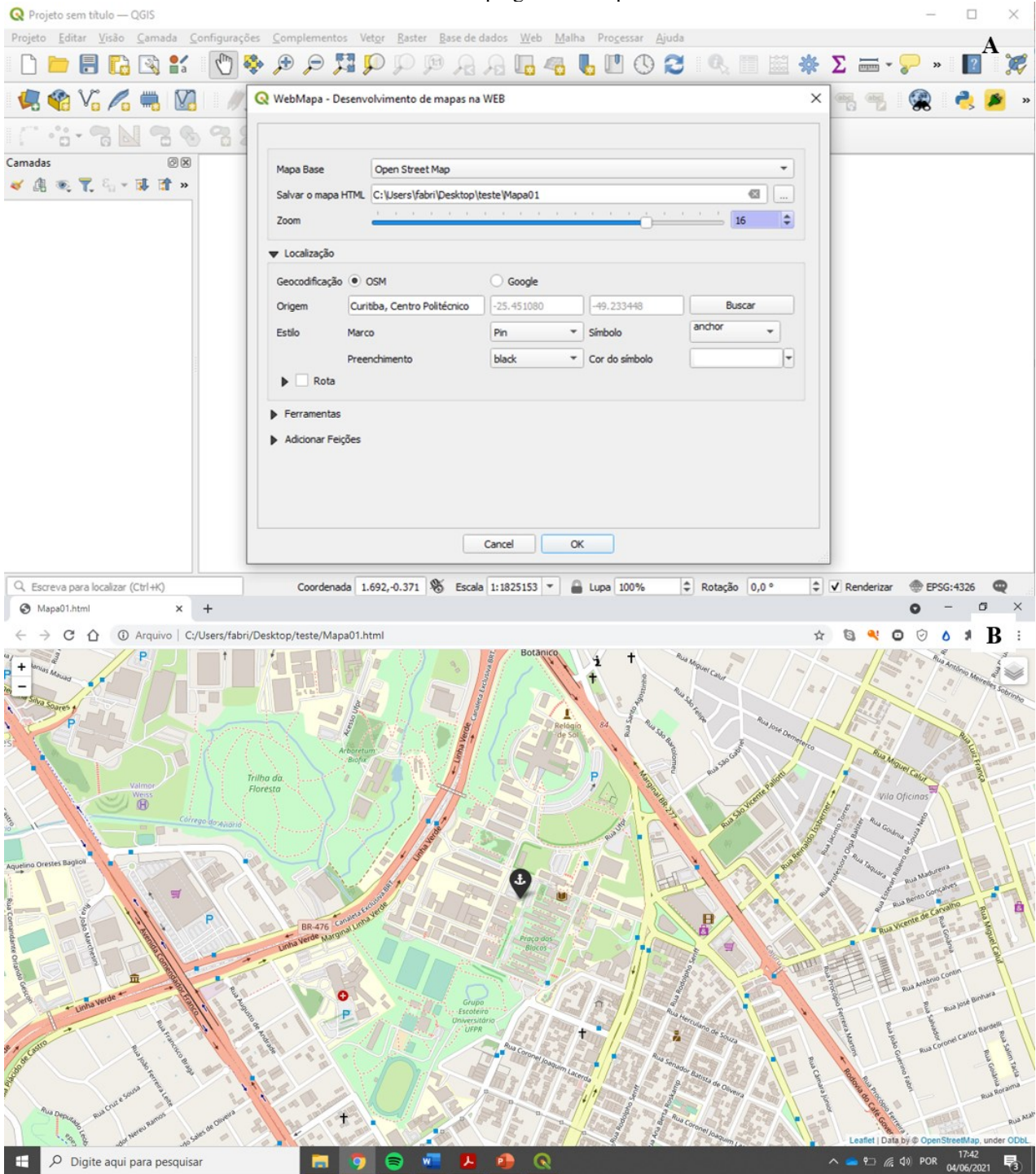
Fonte: CampusMap, 2020.

Para ilustrar a utilização do *plugin* desenvolvido, a experimentação aqui será apresentada em acordo com as seções do WebMapa e seus respectivos resultados em arquivo de formato *.html*. Para tanto, os dados vetoriais da base cartográfica do CampusMap foram inseridos no *plugin* WebMapa. Para realizar as tarefas definidas, empregamos o idioma visual do mapa com a simbolização das feições apresentada na Figura 4. O servidor Web utilizado nos experimentos foi o Google Chrome.

A Figura 5a apresenta a seção inicial do *plugin*, disponível na tela inicial do *software* QGIS, com a seleção da base cartográfica, o nível de zoom do mapa e o ponto central do mesmo. A Figura 5b apresenta o respectivo resultado desse processamento. A Figura 6a, referente à segunda seção do *plugin*, apresenta as opções de incremento de ferramentas cartográficas ao mapa (ex.: escala, minimap), a Figura 6b mostra os resultados iniciais obtidos com o processamento mostrado na Figura 6a. Para apresentar os resultados da terceira seção do *plugin*, a Figura 7a disponibiliza as opções de traçado e simbolização de rotas, enquanto a Figura 7b representa a rota. A Figura 8a mostra a utilização de feições cartográficas vetoriais (ponto, linha e polígono) sobrepostas ao mapa (Figura 8b). As opções para simbolização da rota e das feições (em acordo com a Figura 4) são descritas nas respectivas etapas de processamento mostradas nas figuras.

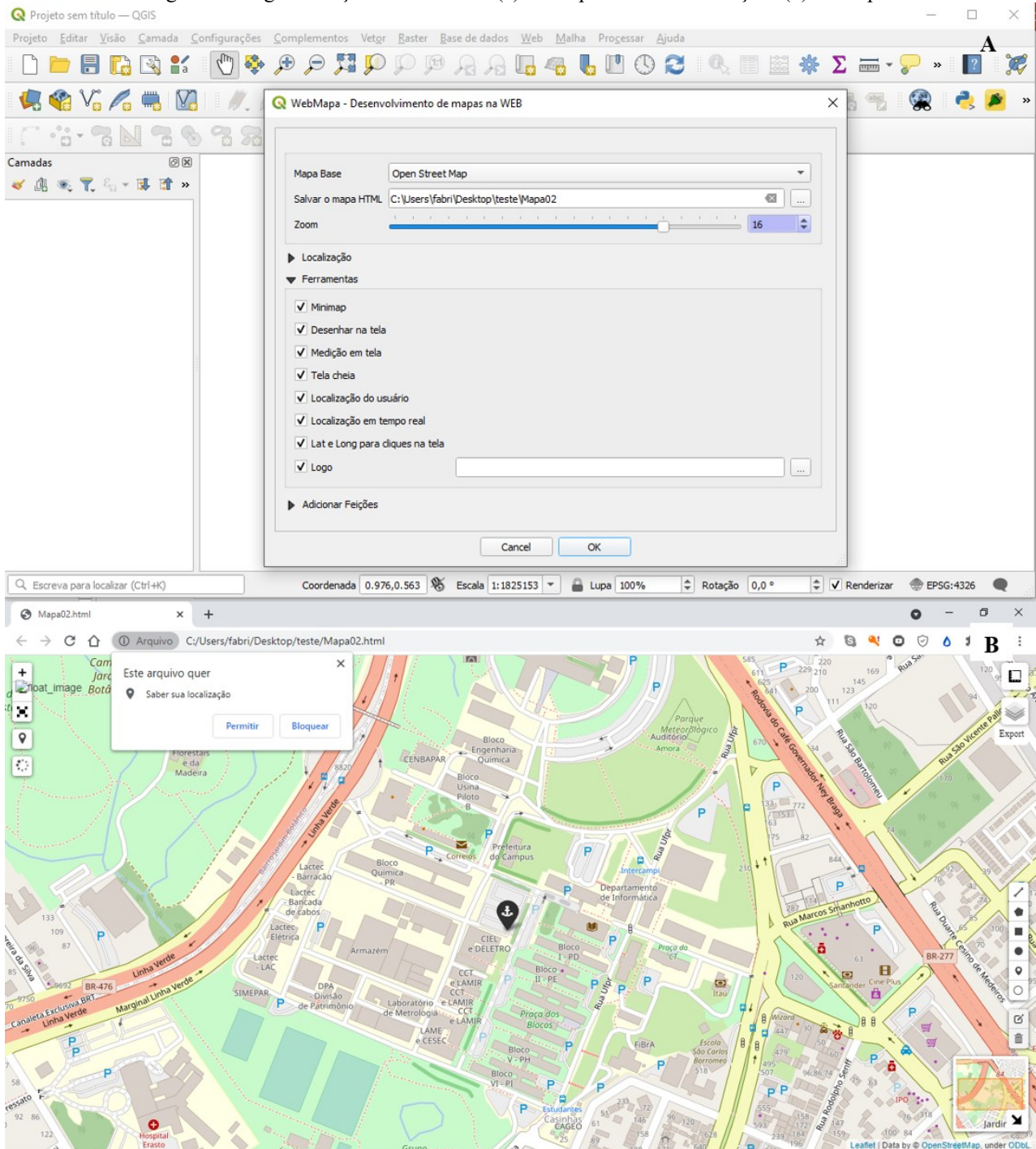
O arquivo final do *plugin* desenvolvido nesse trabalho, denominado “WebMapa”, está disponibilizado no sítio GitHub (AMORIM, 2021a) para *download* como um arquivo compactado. Após o *download*, para instalá-lo no QGIS é necessário acessar a aba Complementos -> Gerenciar e Instalar Complementos -> Instalar a partir do ZIP -> Selecionar o arquivo na pasta fonte -> clicar em “Instalar o complemento”. Após a instalação, o *plugin* “WebMapa” estará disponível na tela principal do QGIS para a sua utilização. A explicação sobre o processo de instalação e utilização do *plugin* WebMapa pode ser encontrada em um vídeo disponível no Youtube (AMORIM, 2021b).

Figura 5 – Interface inicial do plugin WebMapa aberta no *software* QGIS (a) e Resultado do processamento da seção inicial do plugin WebMapa.



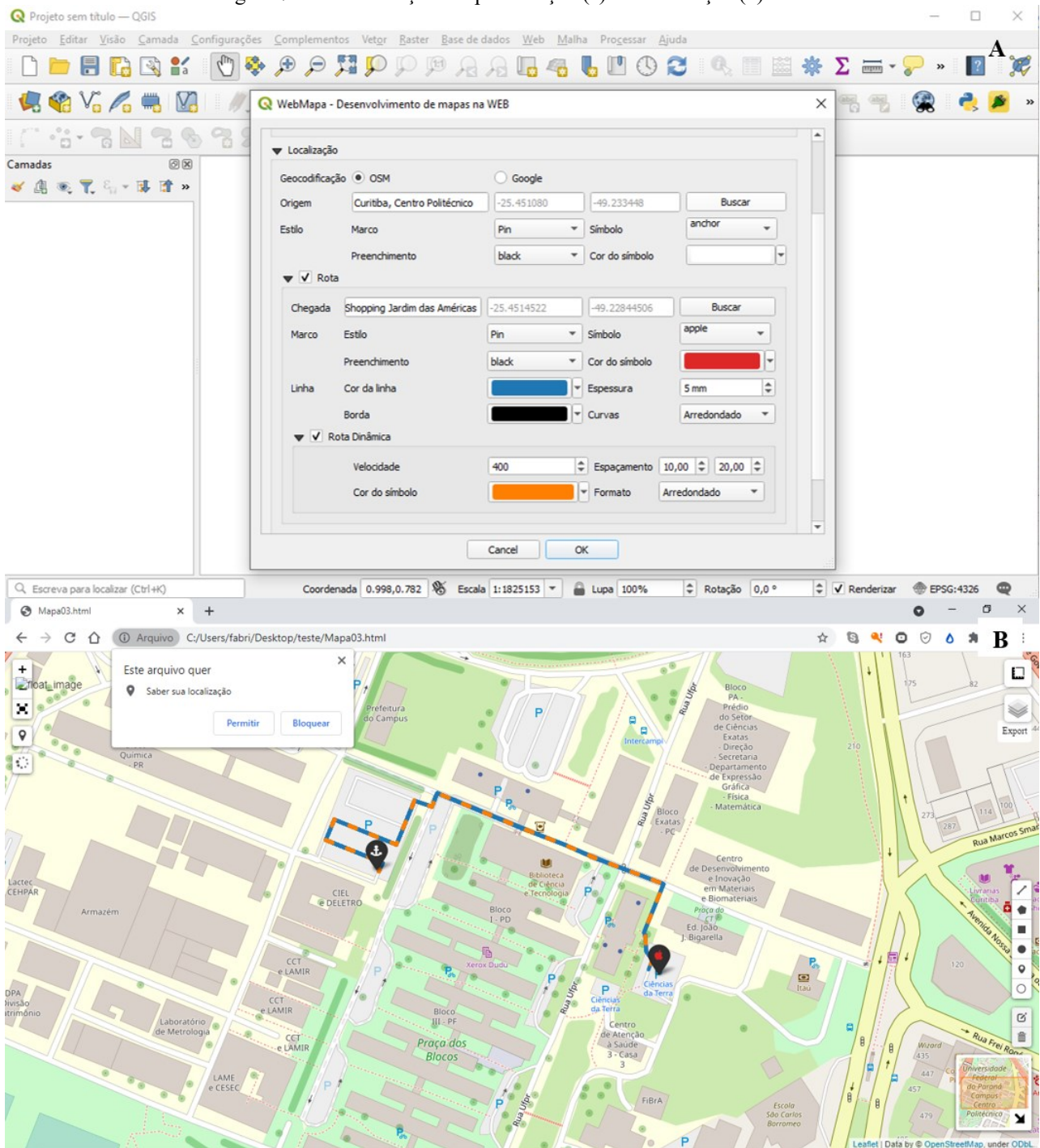
Fonte: Os autores (2021).

Figura 6 – Segunda seção - ferramentas (a) do mapa e suas visualizações (b) no mapa.



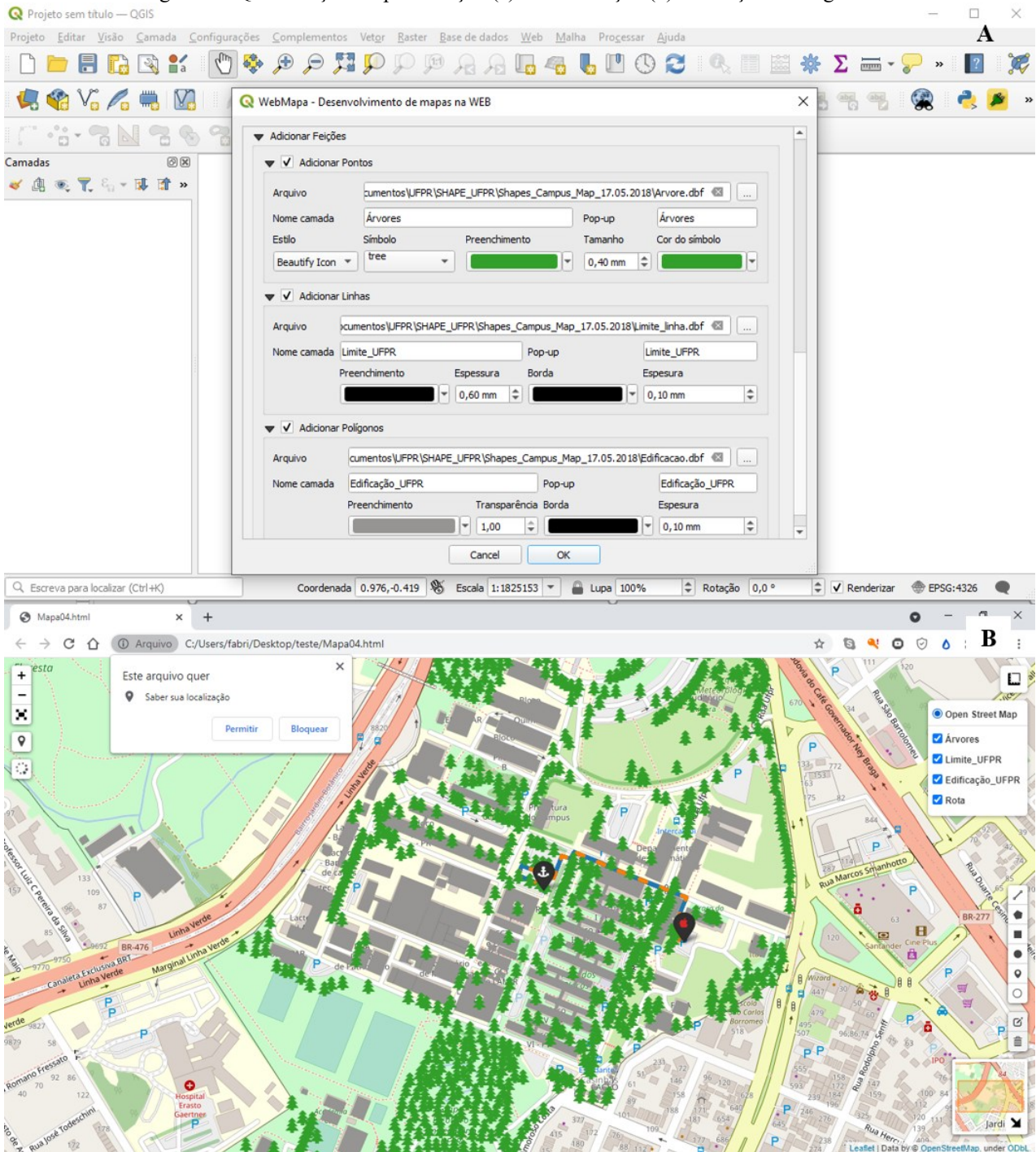
Fonte: Os autores (2021).

Figura 7 – Terceira seção – representação (a) e visualização (b) da rota.



Fonte: Os autores (2021).

Figura 8 – Quarta seção - representação (a) e visualização (b) das feições cartográficas.



Fonte: Os autores (2021).

4 CONSIDERAÇÕES FINAIS

Este artigo apresentou brevemente a possibilidade de criar mapas de interação aberta em navegadores, sob a hipótese de utilização de um conjunto de insumos livres. A biblioteca Folium, em linguagem *python*, foi útil para a manipulação, a análise e a representação dos dados aqui utilizados por meio da base cartográfica do projeto CampusMap, da Universidade Federal do Paraná - UFPR. A utilização de dados, códigos e plataformas livres se mostrou uma alternativa à produção de mapas Web por meio de algoritmos. Por meio dos materiais utilizados nesse trabalho para a produção do *plugin*, percebeu-se a vasta gama de produtos cartográficos que podem ser gerados de maneira efetiva e rápida. Existem diversas bibliotecas *python* que podem ser utilizadas para gerar mapas (ex.: *matplotlib*, *gdal*, *arcpy* e *pandas*), todas elas podem ser utilizadas em conjunto com o projeto aqui desenvolvido e aumentar as opções de visualização de mapas por meio de funções conhecidas como *back-end*. O projeto de interfaces aqui proposto foi executado com o auxílio do *software* QtDesigner, no

entanto, outros *softwares* e métodos poderiam ser utilizados nessa etapa para realizar as funções *front-end*.

A interface gráfica produzida para ser exibida e utilizada no QGIS permite que os usuários selecionem interativamente, naveguem, pesquisem e filtrem eventos de interesse. Os usuários podem visualizar espacialmente a localização do evento escolhido, bem como inspecionar outras propriedades calculadas, como área. Além da visualização interativa, o mapa quando acrescido de dados (camadas) pode ser usado para análises científicas. O *plugin* desenvolvido ainda carece de ajustes, futuras etapas podem complementá-lo, por exemplo: fornecer outras opções para o processamento; e fornecer outras opções de resultados. Futuros trabalhos serão desenvolvidos para aprimorar o trabalho até aqui desenvolvido. Para tal, mais opções de processamento poderão ser acrescidas no sistema. Por exemplo, mais opções de símbolos estáticos e dinâmicos para representar rotas, utilização de um banco de dados cartográficos para manter cada vez mais feições ao mapa (o que requer cuidados estruturais).

Agradecimentos

Ao CNPq pelo apoio financeiro para a realização da pesquisa, processo nº 142253/2020-0.

Referências

- ALAMEH, N. Service chaining of interoperable geographic information Web services. **Internet Computing**, v. 7, n. 1, p. 22-29, 2002.
- AMORIM, A.; DE SOUZA, G. H. B.; YAMASHITA, M. C. Cadastro técnico multifinalitário via internet: um importante instrumento de apoio ao planejamento municipal. **Revista Brasileira de Cartografia**, v. 60, n. 2, 2008.
- AMORIM, F. R. WebMapa: Plugin QGIS para desenvolvimento de mapas online / na Web. **Youtube**, 31 mar. 2021. Disponível em: <https://www.youtube.com/watch?v=sn3BSf9nJIg>. Acesso em: 04 jun. 2021.
- AMORIM, F. R. WebMapa. **GitHub**, 29 mar. 2021. Disponível em: (<https://github.com/CartoFabricio/WebMapa>). Acesso em: 04 jun. 2021.
- CARTWRIGHT, W. Google Maps and mobile devices: Can just one generic design work?. **Revista Brasileira de Cartografia**, v. 60, n. 3, 2008.
- CHOURASIA, A.; RICHARDS-DINGER, K. B.; DIETERICH, J. H.; CUI, Y. Visual Exploration and Analysis of Time Series Earthquake Data. In: **Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact**. p. 1-6, 2017.
- FALAT, D. R.; DELAZARI, L. S. Avaliação de Mapas na Web: Questões relativas à interface e à interatividade. **Revista Brasileira de Cartografia**, v. 62, n. 2, 2010.
- FOLIUM: python data leaflet.js. Retrieved Jun 8, 2017 Disponível em: <https://github.com/python-visualization/folium>. Acesso em: 20 set. 2020.
- FOLIUM. **Visão geral do folium**. Disponível em: <https://python-visualization.github.io/folium/quickstart.html>. Acesso em: 20 set. 2020.
- FOLIUM. **Exemplos de utilização do folium**. Disponível em: <https://github.com/python-visualization/folium/tree/master/examples>. Acesso em: 20 set. 2020.
- GEOJSON, 2016. Disponível em: <http://geojson.org> >. Acesso em: 20 set. 2020.
- ICA. **Commission on Maps and the Internet**. Disponível em: <http://maps.unomaha.edu/ica/>>. Acesso em: 20 set. 2020.
- JAWLA, A.; SINGH, M.; HOODA, N. **Crime Forecasting using Folium**. 2020.
- KRAAK, J.; BROWN, A. **Web cartography**. CRC Press, 2003.
- LEAFLET – a JavaScript library for interactive maps. Disponível em: <http://leafletjs.com> >. Acesso em: 15 set. 2020.
- LEDERMANN, F.; GARTNER, G. Mapmap. js: a data-driven web mapping API for thematic

- cartography. **Revista Brasileira de Cartografia**, v. 67, n. 5, 2015.
- MACEACHREN, A. M.; KRAAK, M. Research challenges in geovisualization. **Cartography and geographic information science**, v. 28, n. 1, p. 3-12, 2001.
- MARISCO, N.; PHILIPS, J.; PEREIRA, H. R. Protótipo de mapa para web interativo: uma abordagem utilizando código aberto. **Revista Brasileira de Cartografia**, v. 56, n. 1, 2004.
- DE MELO JUNIOR, J. B.; CANDEIAS, A. L. B.; JUNIOR, J. R. T. Serviços web geográficos e sua interoperabilidade. **Revista Brasileira de Cartografia**, v. 62, n. 1, 2010.
- DE MELO JUNIOR, J. B.; CANDEIAS, A. L. B. SIG e sua interoperabilidade utilizando servidores de WEB. **Revista Brasileira de Cartografia**, 2005.
- MENEGUETTE, A. A. C. Geovisualização: Exercícios Práticos em Sala de Aula. **Revista Brasileira de Cartografia**, v. 66, n. 4, 2014.
- O'REILLY, T. (2005) **What is web 2.0**: Design Patterns and Business Models for the Next Generation of Software. Disponível em: <<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web20.HTML>>. Acesso em: 20 set. 2020.
- PAPPALARDO, L.; SIMINI, F.; BARLACCHI, G.; PELLUNGRINI, R. **scikit-mobility**: A Python library for the analysis, generation and risk assessment of mobility data. 2019.
- PETERSON, M. P. Evaluating mapping APIs. In: **Modern Trends in Cartography**. Springer, Cham, p. 183-197, 2015.
- PYTHON. Disponível em:<<https://www.python.org>>. Acesso em: 20 set. 2020.
- SAMRETH, D.; ARNAVIELHE, S.; INGENRIETH, F.; BEDBROOK, A.; ONORATO, G. L.; MURRAY, R.; ... & MALVA, J. Geolocation with respect to personal privacy for the Allergy Diary app-a MASK study. **World Allergy Organization Journal**, 11(1), 15, 2018.
- STYLOS, J.; MYERS, B. Mapping the space of API design decisions. In: **IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2007)**. IEEE, p. 50-60, 2007.
- TSOU, M. H. An intelligent software agent architecture for distributed cartographic knowledge bases and Internet mapping services. In: **Maps and the Internet**. Elsevier Science, p. 231-245, 2003.
- VAN ELZAKKER, C. P. J. M. Thinking aloud about exploratory cartography. In: ICC, 1999: Proceedings of the 19th International Cartographic Conference and 11th general assembly: touch the past, visualize the future: August 14-21, Ottawa. **International Cartographic Association (ICA)**, 1999.

Biografia do autor principal



Fabrício Rosa Amorim, nascido em Patos de Minas - MG. Engenheiro Agrimensor e Cartógrafo pela Universidade Federal de Uberlândia – UFU, Campus Monte Carmelo (2015). Mestre em Ciências Cartográficas pela Universidade Estadual Paulista (FCT - UNESP), campus Presidente Prudente (2018). Doutorando em Ciências Geodésicas pela Universidade Federal do Paraná - UFPR. Pesquisador em Percepção visual e espacial para tarefas de navegação apoiadas por mapas digitais estáticos e dinâmicos. Atualmente, pesquisa a percepção visual por símbolos dinâmicos em interfaces de Sistemas de Realidade Aumentada (SRA).



Esta obra está licenciada com uma Licença [Creative Commons Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/) – CC BY. Esta licença permite que outros distribuam, remixem, adaptem e criem a partir do seu trabalho, mesmo para fins comerciais, desde que lhe atribuam o devido crédito pela criação original.