

MAPAS NA WEB E GEOTECNOLOGIAS LIVRES

Elmo Neto

IFSC - Câmpus Garopaba
23 de outubro de 2024



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO

GOVERNO FEDERAL
BRASIL
UNIÃO E RECONSTRUÇÃO

Sobre mim

★ Natural de São Gabriel (RS)

✨ Primeiro contato com informática em 2003

🧑 Instrutor de informática aos 14

gMaps Mapeador do OpenStreetMap desde 2014

🎓 Bacharel em Ciência da Computação pela UFSM

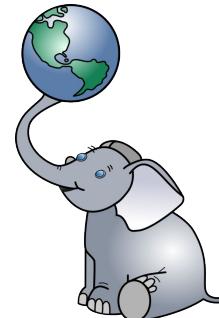
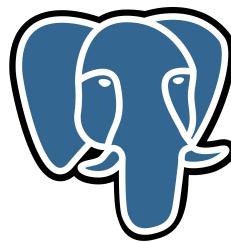
💻 Desenvolvedor de Sistemas de Informações Geográficas

🔍 Pesquisador de Geotecnologias Livres

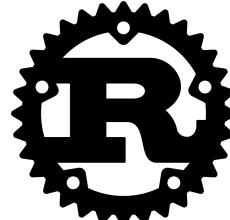


Sobre mim

Trabalho hoje com:



Estudo:



Sobre a apresentação

⌚ 1 hora expositiva + 30 minutos prática

🙋 Levante a mão e interrompa quando quiser

Tópicos:

- Software Livre
- Cartografia Digital
- Geotecnologias Livres
- Mapas na Web
- Prática



SOFTWARE LIVRE

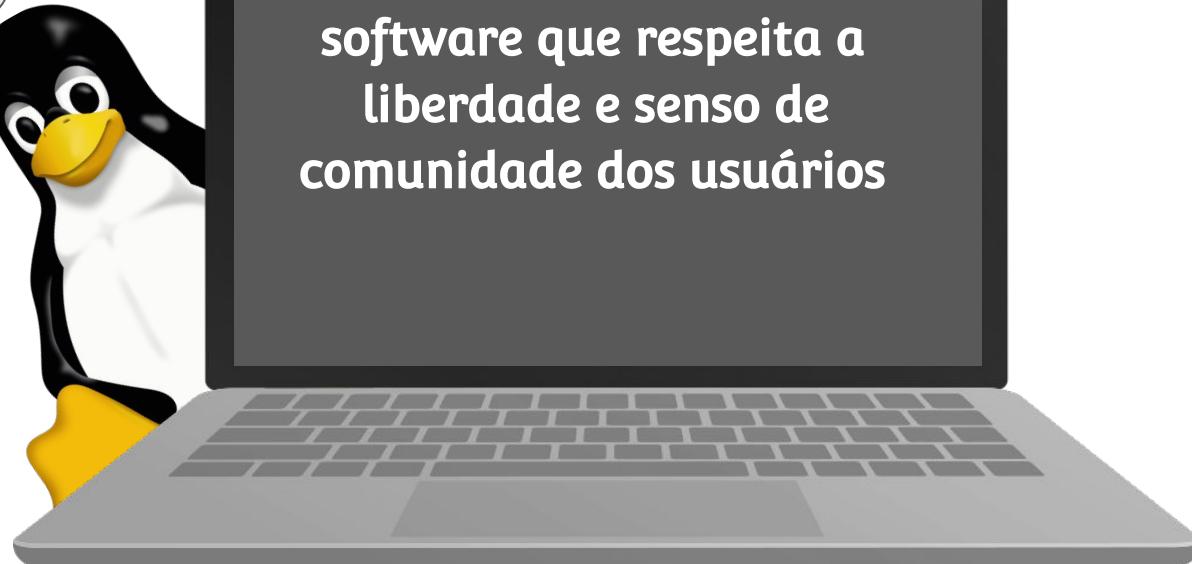
Simon Abrams no Unsplash



Software Livre

Com licença,
já ouvisse a
palavra do
GNU/Linux?

**software que respeita a
liberdade e senso de
comunidade dos usuários**



Liberdades Essenciais

LIBERDADE 0: A liberdade de **executar** o programa como você desejar, para qualquer propósito.

LIBERDADE 1: A liberdade de **estudar** como o programa funciona, e **adaptá-lo** às suas necessidades. Para tanto, **acesso ao código-fonte** é um pré-requisito.

LIBERDADE 2: A liberdade de **redistribuir** cópias de modo que você possa ajudar outros.

LIBERDADE 3: A liberdade de **distribuir** cópias de suas **versões modificadas** a outros

<https://www.gnu.org/philosophy/free-sw.pt-br.html>



Liberdade ≠ Gratuidade

O **free software** muitas vezes é traduzido equivocadamente para **software gratuito**.

Software livre **pode ser gratuito**, mas **não é obrigatoriamente**.

Se o software é livremente modificável e distribuível, **como se cobra** por ele?

- Fornecimento de Infraestrutura
- Treinamentos
- Suporte técnico
- Financiamento Coletivo
- ...



Licenças de software livre

Copyleft

x

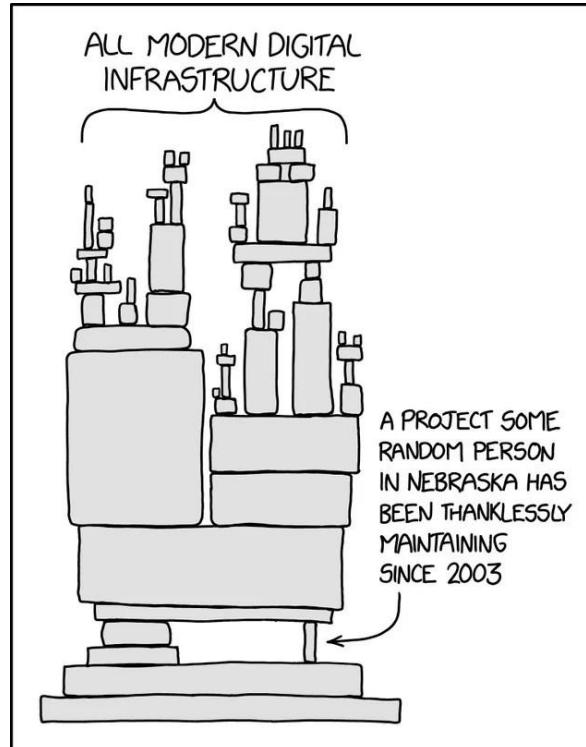
liberdade para executar, modificar e distribuir contanto que cópias originais e modificadas sejam distribuídas nestas mesmas condições

Permissividade

liberdade para executar, modificar e distribuir, incluindo fins comerciais que não mantenham as mesmas liberdades do software original



Licenças de software livre



<https://xkcd.com/2347/>



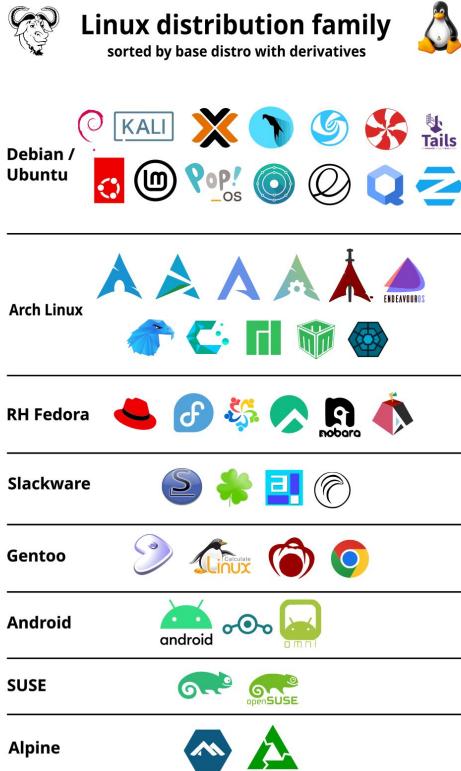
Licenças de software livre

	BSD	MIT	Apache	GPL 2.0	GPLv3	AGPL	LGPL	Mozilla / EPL
Reciprocidade	Ausente	Ausente	Ausente	Total	Total	Total	Parcial	Parcial
Clareza	Média	Alta	Alta	Média	Média	Média	Baixa	Alta
Adoção	Alta	Baixa	Média	Alta	Média	Baixa	Alta	Alta
Compatibilidade com a GPL	Sim	Sim	v3	2.0	v3	Sim	Respectiva	Não
Licenças derivadas	Sim	Sim	Sim	Não	Não	Não	Não	Sim

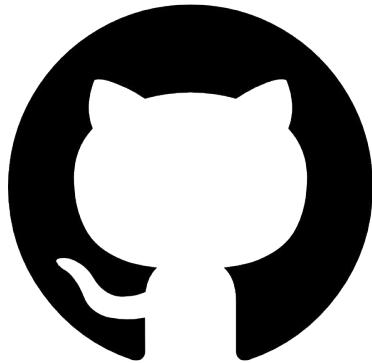
[Artigo na Wikipedia sobre Licenças de Software](#)



Softwares



Hubs de Software Livre



Assistentes de Código

Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions

Hammond Pearce Department of ECE New York University Brooklyn, NY, USA hammond.pearce@nyu.edu	Baleegh Ahmad Department of ECE New York University Brooklyn, NY, USA ba1283@nyu.edu	Benjamin Tan Department of ESE University of Calgary Calgary, Alberta, CA benjamin.tan1@ucalgary.ca	Brendan Dolan-Gavitt Department of CSE New York University Brooklyn, NY, USA brendandg@nyu.edu
---	--	---	--

Abstract—There is burgeoning interest in designing AI-based systems to assist humans in designing computing systems, including tools that automatically generate computer code. The most notable of these comes in the form of the first self-described ‘AI pair programmer’, GitHub Copilot, which is a language model trained over open-source GitHub code. However, code often contains bugs—and so, given the vast quantity of unvetted code that Copilot has processed, it is certain that the language model will have learned from exploitable, buggy code. This raises concerns on the security of Copilot’s code contributions. In this work, we systematically investigate the prevalence and conditions that can cause GitHub Copilot to recommend insecure code. To perform this analysis we prompt Copilot to generate code in scenarios relevant to high-risk cybersecurity weaknesses, e.g. those from MITRE’s “Top 25” Common Weakness Enumeration (CWE) list. We explore Copilot’s performance on three distinct code generation axes—examining how it performs given diversity of weaknesses, diversity of prompts, and diversity of domains. In total, we produce 89 different scenarios for Copilot to complete, producing 1,689 programs. Of these, we found approximately 40 % to be vulnerable.

Index Terms—Cybersecurity, Artificial Intelligence (AI), code generation, Common Weakness Enumerations (CWEs)

I. INTRODUCTION

systematic examination of the security of As GitHub Copilot is the largest and most notable of these comes in the form of the first self-described ‘AI pair programmer’, GitHub Copilot, which is a language model currently available, it is important Copilot’s suggestions commonly insecur prevalence of insecure generated code? “context” yield generated code that is n We systematically experiment with Copilot to these questions by designing scenarios complete, and by analyzing the produced weaknesses. As our corpus of well-defined check Copilot completions for a subset of Weakness Enumerations (CWEs), from Top 25 Most Dangerous Software Weakness list is updated yearly to indicate the most weaknesses as measured over the previous The AI’s documentation recommends this together with testing practices and security your own judgment’. Our work attempts to tendency of Copilot to produce insecure code for the amount of scrutiny a human developer do for security issues.

Assistentes de código são ameaça para a qualidade de software e para o software livre.

An Empirical Evaluation of GitHub Copilot’s Code Suggestions

Nhan Nguyen and Sarah Nadi
University of Alberta
Edmonton, AB, Canada
{nhnguyen,nadi}@ualberta.ca

ABSTRACT

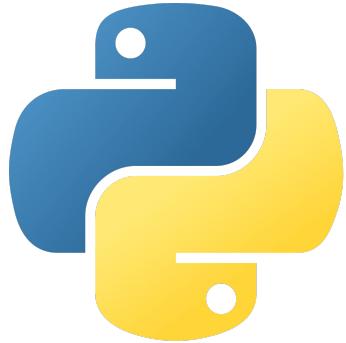
GitHub and OpenAI recently launched Copilot, an “AI pair programmer” that utilizes the power of Natural Language Processing, Static Analysis, Code Synthesis, and Artificial Intelligence. Given a natural language description of the target functionality, Copilot can generate corresponding code in several programming languages. In this paper, we perform an empirical study to evaluate the correctness and understandability of Copilot’s suggested code. We use 33 LeetCode questions to create queries for Copilot in four different programming languages. We evaluate the correctness of the corresponding 132 Copilot solutions by running LeetCode’s provided tests, and evaluate understandability using SonarQube’s cyclomatic complexity and cognitive complexity metrics. We find that Copilot’s Java suggestions have the highest correctness score (57%) while JavaScript is the lowest (27%). Overall, Copilot’s suggestions have low complexity with no notable differences between the programming languages. We also find some potential Copilot shortcomings, such as generating code that can be further simplified and code that relies on undefined helper methods.

Neovim, and JetBrains IDEs [11]. Powered by the large-scale OpenAI Codex model, which was trained on open-source GitHub code, Copilot can suggest code snippets in different programming languages [11]. While there have been lots of similar research efforts [12, 14, 31], the seamless integration and availability of Copilot, along with GitHub’s buzzword, naturally created a “hype” in the tech world with many developers already using it through its technical preview or awaiting its usage [1, 21]. However, as pointed by GitHub, Copilot’s suggestions “may not always work, or even make sense” [11]. Thus, it is important to assess the correctness and quality of Copilot’s suggestions to provide better insights into the overall performance of the tool.

This paper contributes an empirical assessment of GitHub Copilot’s capabilities. The insights gained from such a study can help developers understand how to best use Copilot as well as provide insights to Copilot’s team and other researchers in this area. We specifically focus on Copilot’s synthesized code suggestions, given a natural language description, and answer the following research questions: *RQ1: How correct are Copilot’s code suggestions?* and *RQ2: How understandable is the code provided by Copilot?*



Python



Python ajuda a democratizar o desenvolvimento de software



Oct 2024	Oct 2023	Change	Programming Language	Ratings	Change
1	1		Python	21.90%	+7.08%
2	3	▲	C++	11.60%	+0.93%
3	4	▲	Java	10.51%	+1.59%
4	2	▼	C	8.38%	-3.70%
5	5		C#	5.62%	-2.09%
6	6		JavaScript	3.54%	+0.64%



Governança

Governo se beneficia pela priorização de software livre

- Soberania x colonialismo
- Proteção de dados
- Custo de manutenção
- Extensibilidade



CARTOGRAFIA DIGITAL

Maël BALLAND no Unsplash

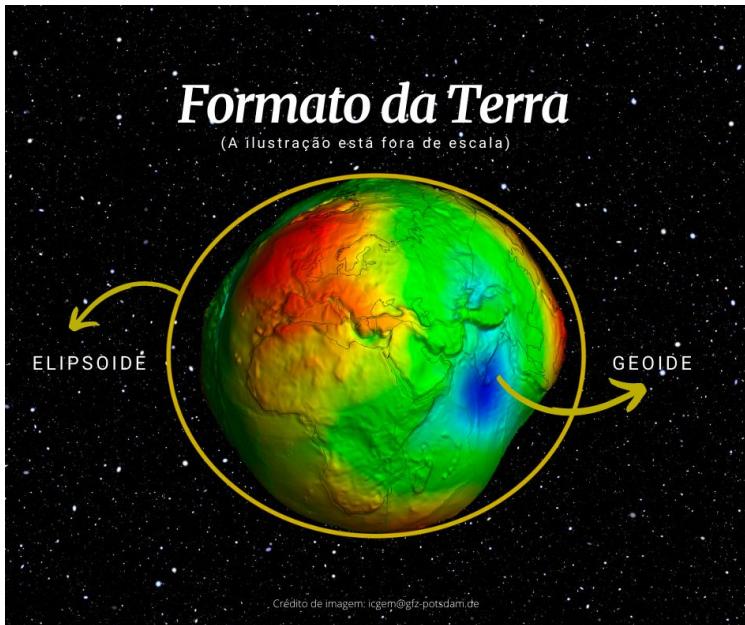


Conceitos Importantes

- Representações da Terra
- Coordenadas Geográficas
- Sistemas de Referências de Coordenadas
- Tipos de Arquivos
- Formatos de Arquivos
- Representação de Dados Vetoriais
- Tiling



Representação da Terra

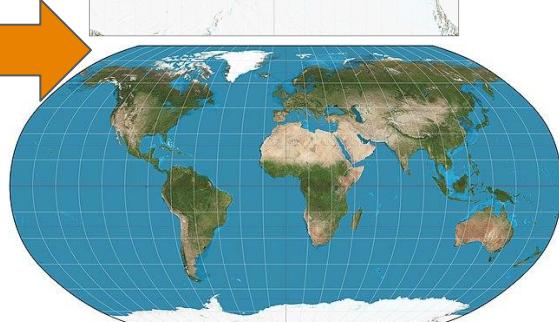


<http://apoiogeomatica.com.br/blog/tag/planeta/>

A projeção de Mercator ainda é muito utilizada na Web, apesar de distorcer consideravelmente porções de terra do Hemisfério Norte. Hoje há modelos híbridos que combinam projeção de Robinson a nível de visão do planeta e projeção de Mercator em nível de zoom para um continente específico.



Projeção de Mercator



Projeção de Robinson

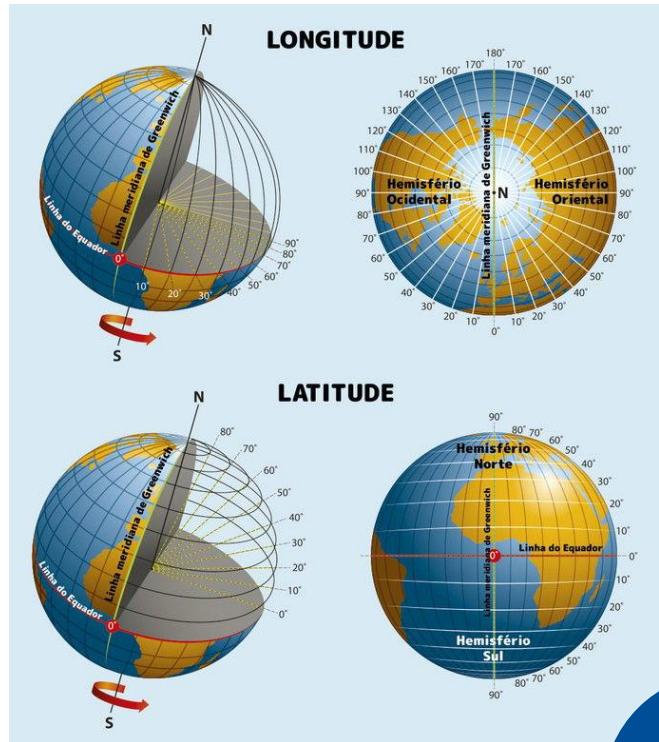


Coordenadas Geográficas

Latitude varia de -90 a 90 ou 90 S a 90 N
Longitude varia de -180 a 180 ou 180 O a 180 L

Cuidado: estamos acostumados a pensar no par **latitude, longitude** que não são **x, y**, mas sim **y, x**

Na prática, cada tecnologia define qual das duas informações deve ser informada primeiro (documentação)



<https://www.todamateria.com.br/latitude-e-longitude/>



Coordenadas Geográficas

WHAT THE NUMBER OF DIGITS IN YOUR COORDINATES MEANS	
LAT/LON PRECISION	MEANING
28°N, 80°W	YOU'RE PROBABLY DOING SOMETHING SPACE-RELATED
28.5°N, 80.6°W	YOU'RE POINTING OUT A SPECIFIC CITY
28.52°N, 80.68°W	YOU'RE POINTING OUT A NEIGHBORHOOD
28.523°N, 80.683°W	YOU'RE POINTING OUT A SPECIFIC SUBURBAN CUL-DE-SAC
28.5234°N, 80.6830°W	YOU'RE POINTING TO A PARTICULAR CORNER OF A HOUSE
28.52345°N, 80.68309°W	YOU'RE POINTING TO A SPECIFIC PERSON IN A ROOM, BUT SINCE YOU DIDN'T INCLUDE DATUM INFORMATION, WE CAN'T TELL WHO
28.5234571°N, 80.6830941°W	YOU'RE POINTING TO WALDO ON A PAGE
28.523457182°N, 80.683094159°W	"HEY, CHECK OUT THIS SPECIFIC SAND GRAIN!"
28.523457182818284°N, 80.683094159265358°W	EITHER YOU'RE HANDING OUT RAW FLOATING POINT VARIABLES, OR YOU'VE BUILT A DATABASE TO TRACK INDIVIDUAL ATOMS. IN EITHER CASE, PLEASE STOP.

<https://xkcd.com/2170/>



Sistemas de Referência de Coordenadas

Sistemas mais utilizados em webmapas:

EPSG:4326 (graus)

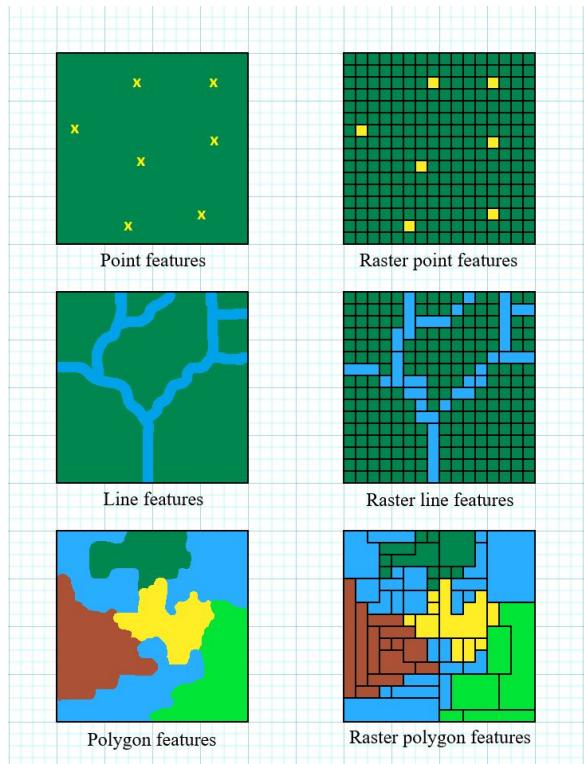
EPSG:3857 (metros)

“Números mágicos” que de tanto você usar vai acabar decorando

Outros sistemas mais apropriados para determinadas regiões podem ser usados, dependendo do escopo do projeto

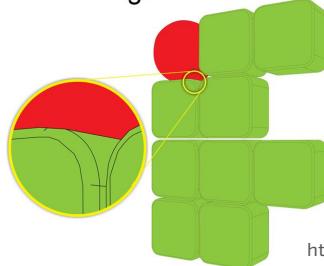


Tipos de Arquivos



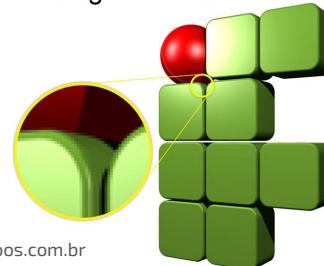
[Trabalho no ResearchGate](#)

Imagen Vector



<https://www.quoos.com.br>

Imagen Raster



Vetor

formado por pontos, linhas,
expressões matemáticas

Natureza discreta

Sistema viário, edificações etc

Maior custo computacional

Menor espaço em disco

Raster

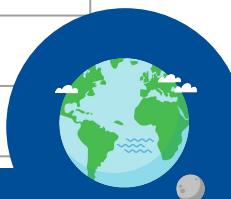
Matriz de pixels com valor de cor

Natureza Contínua

Elevação, cobertura do solo etc

Menor custo computacional

Maior espaço em disco



Formatos de Arquivos

Vetoriais



Shapefile

, criado pela ESRI
Ultrapassado por vários motivos, o principal é que o formato completo é formado por arquivo principal e mais 3 auxiliares



Keyhole Markup Language

Formato padrão do Google Earth
Carrega muitas informações desnecessárias



GeoJSON

Basicamente um JSON com várias features (desenhos) descritas em suas coordenadas e metadados de tipo de geometria e sistema de coordenadas
O preferido para trabalho com dados geoespaciais na Web



GeoPackage

Formato relativamente novo que corrige os problemas do Shapefile. Por baixo dos panos, é um arquivo SQLite com algumas operações espaciais embutidas.

Matriciais



Raster

Principal formato de dados raster
Um pixel tem uma coordenada associada a ele



ECW

Formato proprietário que oferece maior nível de compactação, mas tem o contraponto de ser proprietário e ter processo de conversão dificultado por instalação de SDKs e recompilação de GDAL no Linux



Representação de Dados Vetoriais

Formato Plano (WKT) - fácil leitura por humanos e baixo custo computacional, mas tamanho proibitivo.

Formato Binário (WKB) - difícil leitura por humanos e moderado custo computacional, independente de arquitetura de máquina.

Exemplo com geometria da Rua Maria Aparecida Barbosa em frente ao IFSC:

WKT - Well Known Text:

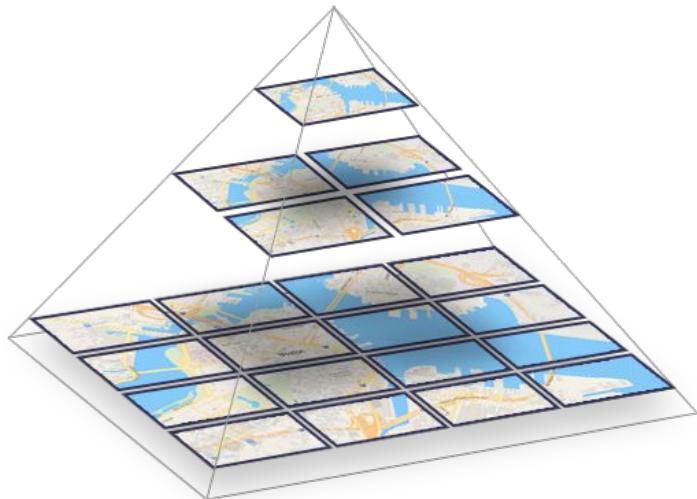
```
LINESTRING(-48.6736499 -28.0993459,-48.674119431915095  
-28.09898906756934,-48.67423231758925  
-28.098897043610233,-48.674415298593004  
-28.098778320395745,-48.6756469055669 -28.098005127300645)
```

WKB - Well Known Binary :

```
0102000020E61000000500000012BAF0283A5648C0DF7D9FBB6E193CC033D6A  
88B495648C032DFF75857193CC0DFF99C3E4D5648C01FFA0F5151193CC084B09  
13D535648C04EA7378949193CC0D78609997B5648C0BA9830DD16193CC0
```



Tiling



<https://docs.maptiler.com/google-maps-coordinates-tile-bounds-projection/>

Mosaico de imagens de mesma dimensão (256x256, 512x512 etc)

Níveis de zoom são carregados conforme demanda

Mais comum são em formato raster, mas hoje vetores também são adotados

Cada nível de zoom tem o quádruplo de imagens do nível anterior (mais distante)



Open Geospatial Consortium (OGC)

Consórcio que discute e publica padrões de interoperabilidade de dados e operações geoespaciais

OGC APIs

Standards that build upon the legacy of the OGC Web Service Standards but define resource-centric APIs that take advantage of modern web development practices.

- OGC API – Common
- GeoAPI Implementation Specification
- OGC API – Tiles

- OGC API – Environmental Data Retrieval
- OGC API – Maps
- OGC SensorThings API

- OGC API – Features
- OGC API – Processes

Services

Standards that implement XML Remote Procedure Calls using the HyperText Transfer Protocol (HTTP).

- 3D Portrayal Service
- Web Services Security
- Web Coverage Service
- Web Map Tile Service
- Web Map Service

- Coordinate Transformation Service
- Table Joining Service
- Web Feature Service
- Web Processing Service

- Location Service (OpenLS)
- Web Coverage Processing Service (WCPS)
- Web Map Context
- Web Service Common

Data Models and Encodings — General

Standards that provide general rules to organize geospatial information, typically sent by a service provider or produced by an application.

- OGC GeoTIFF
- EO Dataset Metadata GeoJSON(LD)
- Geography Markup Language (GML)
- Geospatial User Feedback (GUF)
- KML
- OGC Open Modelling Interface (OpenMI)
- Simple Features for CORBA
- Styled Layer Descriptor
- OGC Two Dimensional Tile Matrix Set

- CoverageJSON
- Filter Encoding
- GML in JPEG 2000 for Geographic Imagery
- GeoSPARQL – A Geographic Query Language for RDF Data
- OGC Moving Features
- Open GeoSMS – Core
- Simple Features for OLE/COM
- Symbology Encoding
- Well-known text representation of coordinate reference systems

- Common Query Language (CQL2)
- Geodetic data Grid eXchange Format (GGXF)
- GeoPose
- GeoXACML
- Observations, Measurements, and Samples
- Simple Feature Access – Part 1: Common Architecture
- Simple Feature Access – Part 2: SQL Option
- Training Data Markup Language for Artificial Intelligence
- Symbology Conceptual Core Model



GEOTECNOLOGIAS LIVRES

NASA no Unsplash



Definição

As geotecnologias são o conjunto de tecnologias para **coleta, processamento, análise e oferta** de informação com referência geográfica (ROSA, 2011)



Áreas

- sistemas de informação geográfica (SIGs)
- cartografia digital
- sensoriamento remoto
- sistema de posicionamento global
- ...





Sistemas de Informações Geográficas (SIGs) realizam o tratamento computacional de dados geográficos e recuperam informações não apenas com base em suas características alfanuméricas, mas também através de sua localização espacial.

coleta • processamento • armazenamento • distribuição • visualização

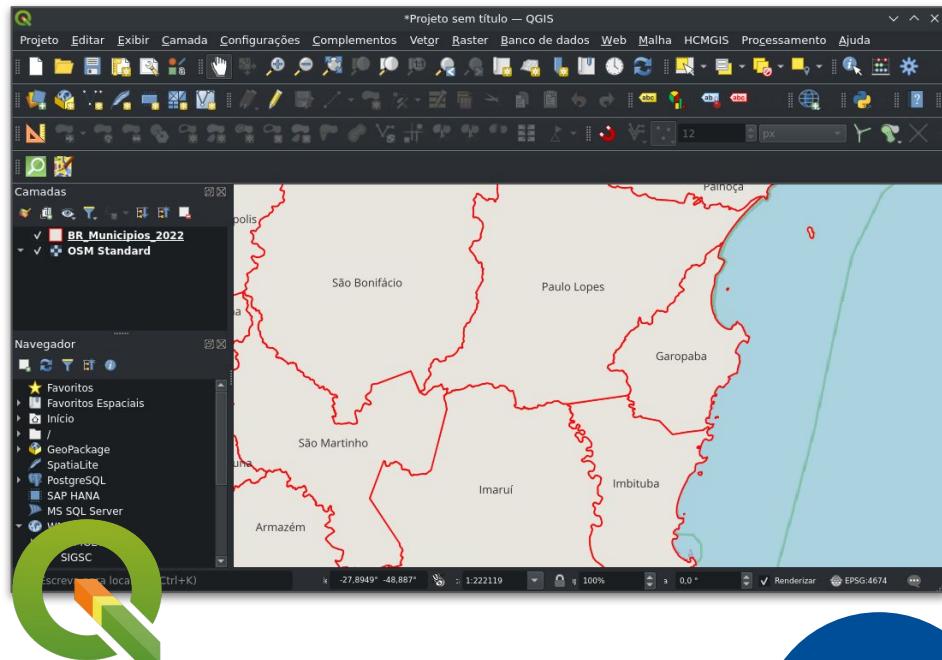




SIG que segue as premissas de **software livre** e que revolucionou a área de estudo até então monopolizada pelo ArcGIS da ESRI.

Hoje o campo de **geociências abertas** e reproduzíveis tem ele como referência.

Instalado nos laboratórios do IFSC!

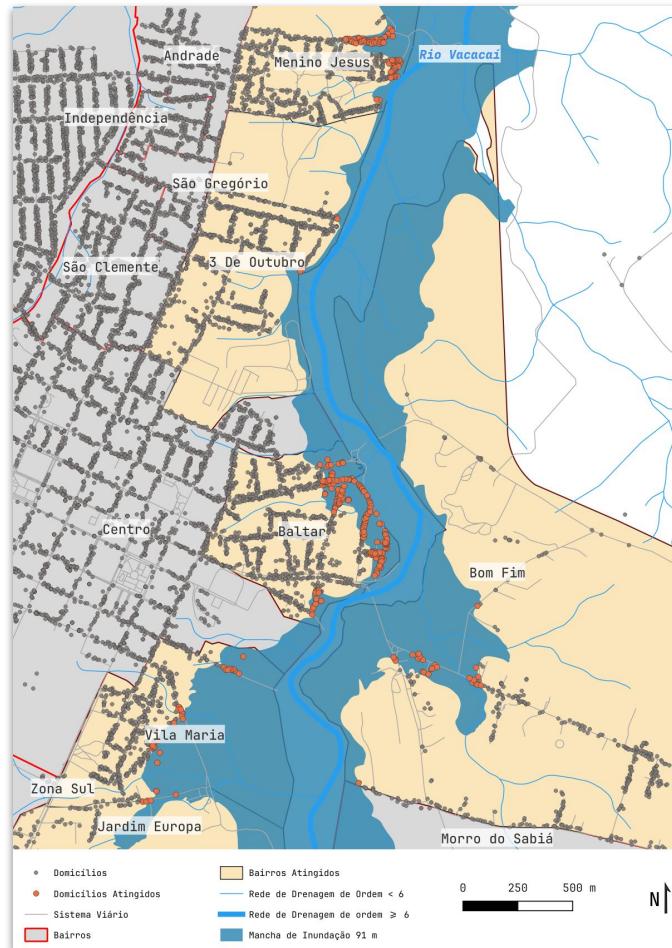




Análise da mancha de inundação decorrente das chuvas de maio de 2024 em São Gabriel, RS



<https://bussola.elmoneto.net/2024/05/geoprocessamento-na-resposta-a-eventos-climaticos-extremos/>



OpenStreetMap



É um projeto colaborativo que tem como objetivo a criação de uma base cartográfica de livre utilização recobrindo todo o planeta.

Qualquer pessoa pode colaborar com edições no mapa e essas edições podem servir a qualquer fim a qualquer outra pessoa ou organização, inclusive com propósitos comerciais.

Frequentemente citado como uma "**Wikipedia de mapas**"



OpenStreetMap

Enquanto a base cartográfica do Google é proprietária e impõe diversas restrições por meio de uma licença rígida de acesso e utilização dos dados brutos, a licença do OpenStreetMap é ODbL (Open Database License), permitindo **livre acesso** e **utilização os dados**, com objetivo comercial ou não, tendo como condições:

- 1) o crédito aos colaboradores do projeto no seu trabalho;
- 2) o banco de dados baseado no e derivado do OSM deve ser disponibilizado nos mesmos termos da ODbL.

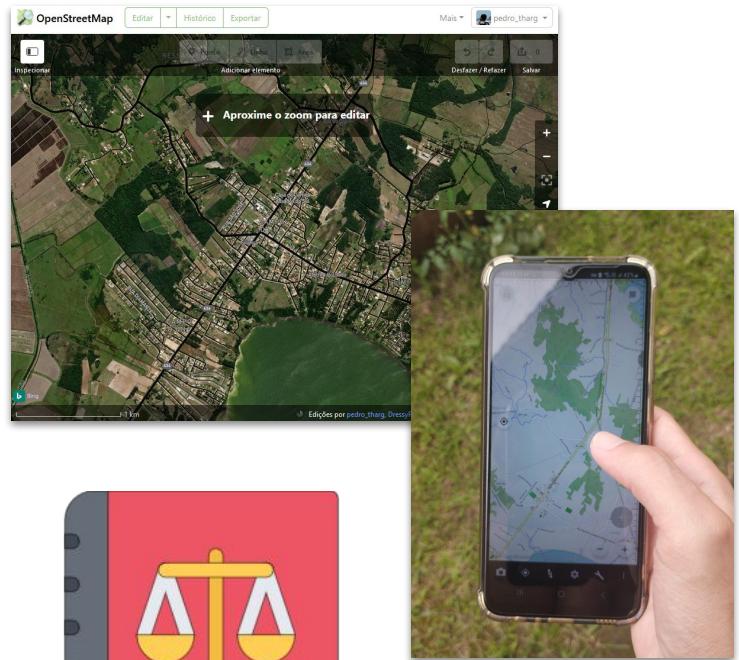


OpenStreetMap

Os dados do OSM são mapeados com base em três fontes principais: **imagens de satélite, coleta de dados em campo e legislação.**

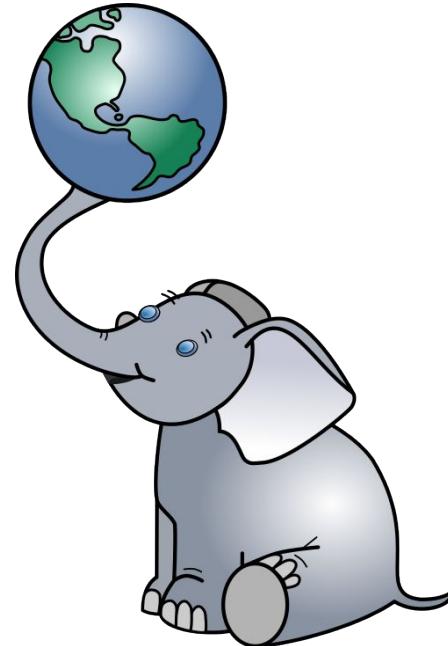
Mapeadores monitoram regiões específicas e ajudam a manter a qualidade do que é adicionado ou alterado, além de identificar potenciais atos de vandalismo.

Ferramentas automatizadas fazem análise de qualidade e identificam edições suspeitas ou incorretas.



PostGIS

Extensão para o PostgreSQL que permite armazenar e manipular dados de natureza espacial e geográfica em relações do banco de dados juntamente com outras colunas de tipos simples. Oferece dezenas de funções espaciais.



Ecossistema Front-End

Geospatial Application Development Ecosystem

Map Libraries



Visualization



Analysis



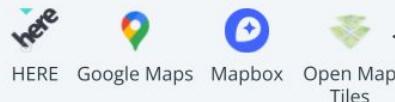
Map Services



Frameworks



Location Data Services



Non-geospatial Frontend Tools



[forrest.nyc](#) [mbforr](#)

geralmente plataformas que oferecem mapas-base para aplicações ou serviços como geocodificação, roteirização etc tem uma cota de uso gratuito e exigem planos pagos em caso de limite de cota ultrapassado



Ecossistema Back-End

Geospatial Python/Spatial Data Science Ecosystem

Analysis/Modeling



Visualization



Geocoding



Geospatial Toolkits



Earth Observation



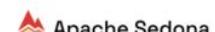
Network Analysis



Non-geospatial Data Science Tools



forrest.nyc [mbforn/](https://www.linkedin.com/in/mbforn/)



Eu preciso mesmo saber tudo isso?

Não



Dê pequenos passos



OpenLayers

Procure tutoriais e documentações de cada projeto e comece pequeno, criando mapas simples.

<https://leafletjs.com/examples.html>

<https://openlayers.org/doc/tutorials/>



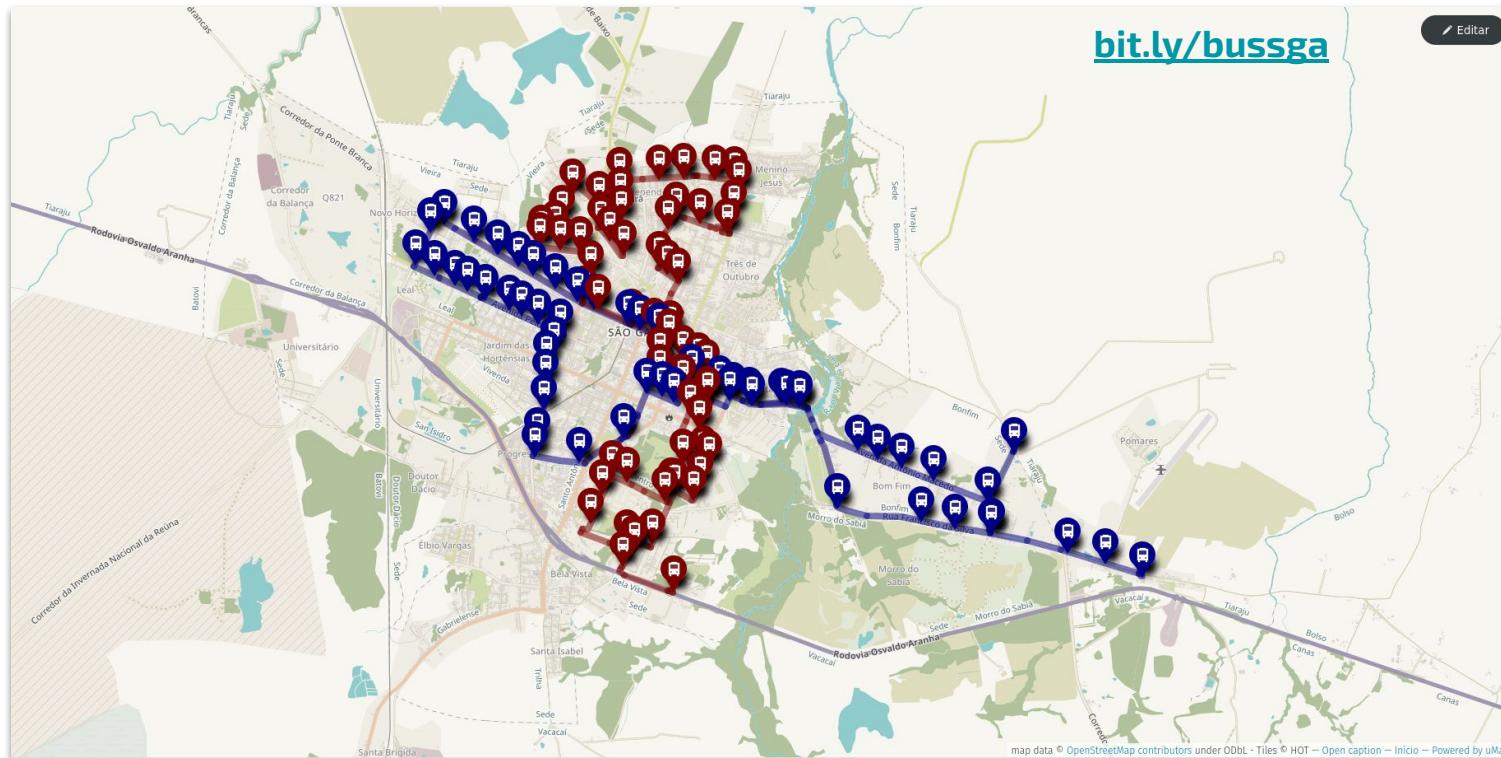
WebMap ≠ WebGIS



MAPAS NA WEB

Patrick Assalé no Unsplash

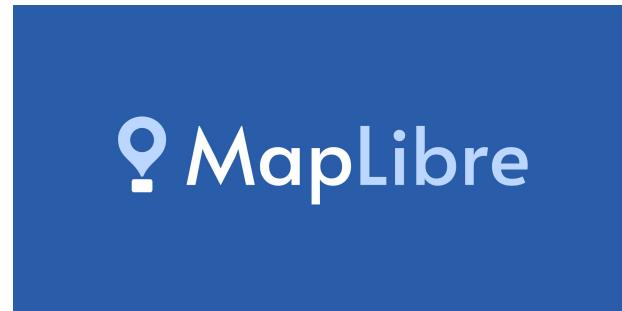




Bibliotecas



OpenLayers



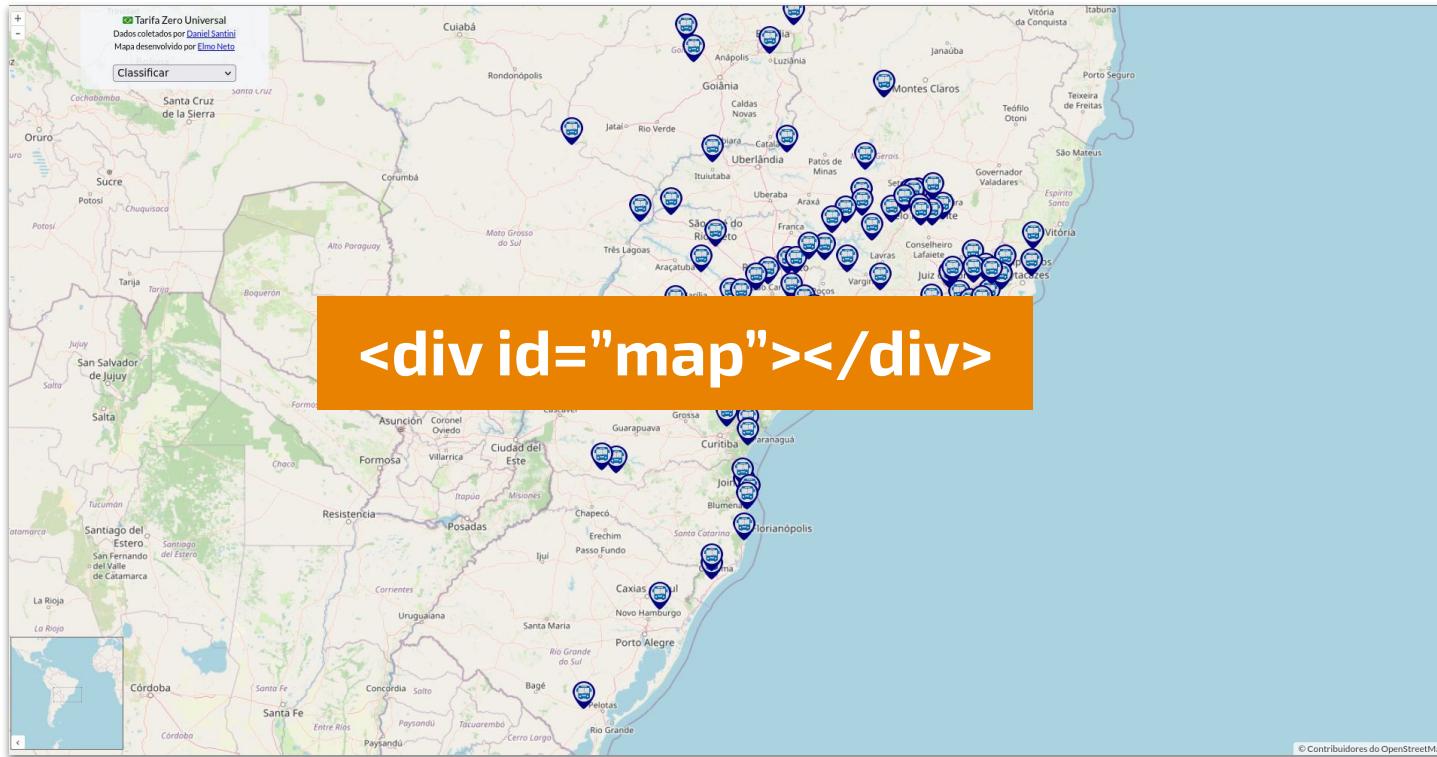
Fornecedores de Tiles



OpenStreetMap



Estrutura de Desenvolvimento



Estrutura de Desenvolvimento

- Importação de JS e CSS da biblioteca de mapas
- Criação de container HTML que será vinculado ao objeto mapa no Javascript
- Criação do objeto mapa no Javascript
- Criação do objeto source/layer
- Customização de controles (opcional)
- Customização de interação (opcional)



Dicas

Escolha o simples

Se você pretende criar uma página simples de visualização de dados e algumas interações, não há necessidade de framework.

Se você pretende criar WebGIS robusto, escolha frameworks e bibliotecas mais adequados ao seu projeto.

Nada de bibliotecas pesadas em tamanho e performance para resolver problema simples.



Dicas

Pense em vários casos

Os dados que você escolheu pra testar o seu mapa podem funcionar bem porque são o que o seu código espera. **Mas e se a estrutura dos dados mudar devido a requisitos de localização geográfica?**

Ex.: sua aplicação lê horários de ônibus em dois cenários: dias úteis e fins de semana. Mas ela está preparada para ler no cenário de feriados? E se sábado e domingo tiverem horários diferentes?



Dicas

Desempenho sobre funcionalidade

Mapas que

- foram feitos para consulta rápida de alguma informação
- são utilizados no deslocamento de algum lugar a outro
- são utilizados em locais onde pode não haver uma boa conexão

Não devem implementar funcionalidades pesadas que exijam muito do dispositivo e da rede.



Antonio Groß no Unsplash



Dicas

Segurança dos Dados

Não envie ao cliente todos os dados vetoriais do mapa de uma só vez.

Envie raster tiles apenas com geometria e estilo.

Caso quem solicita a informação esteja autenticado e tenha acesso, envie dados geométricos vetoriais e demais dados sensíveis.





PRÁTICA

Daniel McCullough no Unsplash



MUITO OBRIGADO

elmo@inf.ufsm.br

t.me/elmoneto

elmoneto.net

