

Ambiente de desenvolvimento

Disciplina: Programação aplicada à engenharia cartográfica

Maurício C. M. de Paulo - D.Sc.

3 de fevereiro de 2026

Instalação do Python (com ambientes virtuais)

Ambiente virtuais em Python:

- Isolamento de dependências entre projetos
- Evita conflitos de versões de bibliotecas
- Permite usar diferentes versões do Python no mesmo sistema
- Facilita a reprodução do ambiente em outros computadores
- Reduz riscos de quebrar aplicações existentes ao instalar novos pacotes

Principal objetivo: instalar GDAL e QGIS

Instalação do Python (com ambientes virtuais)

Gerenciadores de ambientes:

Ferramenta	Descrição	Ambiente	Observações
pip + venv	Pacotes Python	Virtual no projeto	Padrão do Python
uv	pip + venv rápido	Ambientes e pro- jetos	Alto desempenho
Conda	Pacotes diver- sos	Estrutura Conda	Ciência de dados
Pixi	Alternativa ao Conda	Estilo Conda	Substituto mo- derno

- GDAL e QGIS: Pixi (recomendado) ou miniforge (conda).
- Deep learning: uv para pytorch, tensorflow, etc.

Integrated Development Environment (IDE) VSCodeium:

<https://vsodium.com/>

Versão **código aberto** do VS Code.
Usaremos para:

- Escrever código Python
- Depuração (Debug)
- Controle de versões de código (Git)
- Cadernos Jupyter
- Desenhar diagramas
 - Fluxogramas
 - UML: Classes

Quais são as melhores IDE para Python? Forneça também os links para o site do projeto.

Existem várias IDEs para Python, cada uma com suas próprias características e vantagens. Aqui estão algumas das melhores opções:

IDEs Comerciais e Gratuitas

- **PyCharm:** Uma IDE completa com recursos como destaque de sintaxe, preenchimento automático, depuração e gerenciamento de projetos. Disponível em versões comunitária, educacional e profissional. ¹
- Site: (link unavailable)
- **Visual Studio Code (VS Code):** Um editor de código leve e extensível, com suporte a diversas linguagens, incluindo Python. Oferece recursos como depuração, gerenciamento de código e extensões. ¹
- Site: (link unavailable)
- **Spyder:** Uma IDE gratuita e de código aberto, com recursos como destaque de sintaxe, preenchimento automático, depuração e visualização de dados. ¹
- Site: (link unavailable)

IDEs para Desenvolvimento Web e Ciência de Dados

- **Jupyter Notebook:** Uma ferramenta de código aberto para ciência de dados e aprendizado de máquina, com suporte a notebooks interativos. ¹
- Site: (link unavailable)
- **Google Colab:** Um serviço de nuvem gratuito para desenvolvimento de machine learning e inteligência artificial, com suporte a notebooks interativos. ¹
- Site: (link unavailable)

Outras Opções

- **IDLE:** Uma IDE básica e multiplataforma, com recursos como destaque de sintaxe e depuração. ¹
- Site: (link unavailable)

Caso ainda não tenha instalado, essa é a hora de instalar o VSCodium

- Windows: `winget VSCodium.VSCodium`
- Linux: Baixar .deb em
<https://github.com/VSCodium/vscodium/releases>

Copilotos de desenvolvimento

Origem: StackOverflow <https://stackoverflow.com/>

- ChatGPT <https://chatgpt.com/> (gratuito)
- Github Copilot <https://github.com/features/copilot> (gratuito)
- Gemini <https://gemini.google.com> (gratuito)
- Llama: <https://www.meta.ai/> (gratuito + pesos e código abertos)
- **Instalação local:** OLLAMA <https://ollama.com/>

Entre outros, alguns especialistas.

Não confie cegamente no trabalho de outra pessoa.

Muito menos de uma IA.

Extensões disponíveis no VSCode/VSCodium:

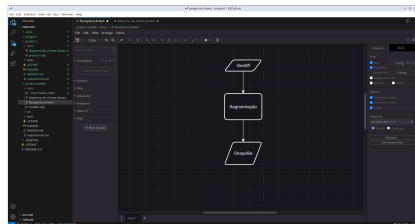
- Github Copilot (nativo)
- Gemini
- Continue.dev + Ollama **Instalação local**
- Codegpt

Cuidados:

- Seu código é compartilhado com o dono do serviço (exceto se o serviço for seu).
- É fácil consumir todos os limites de uso das versões gratuitas.

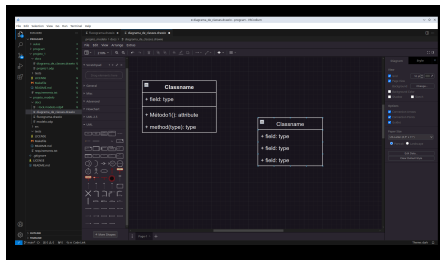
Fluxograma

- Pasta: projeto/docs/fluxograma.drawio
- Abrir com:
<https://www.drawio.com/>
- VSCode Draw.io Integration (Author: Hediet)
- Tutorial de edição: <https://www.drawio.com/doc/getting-started-basic-flow-chart>



Diagramas de classes (UML)

- Pasta:
projeto/docs/diagrama_
de_classes.drawio
- Abrir com:
<https://www.drawio.com/>
- VSCode Draw.io Integration
(Author: Hediet)
- Tutorial de edição:
<https://www.drawio.com/blog/uml-class-diagramss>



- Desenvolvimento ágil em ciclos de 2 semanas
- `https://ronjeffries.com/xprog/what-is-extreme-programming/`
- Programação em pares (Xtreme programming - XP)

- Plugin Hello Map
- Carregar dados matriciais e vetoriais no QGIS (GDAL opcional)
- Abrir coordenadas da rede de monitoramento SIRGAS e converter para um arquivo vetorial
- Baixar e manipular dados tabulados com Pandas e escrever com Geopandas.
- Processing Plugin de extrair cotas de um Modelo Digital de Superfícies
- Manipular dados matriciais e vetoriais no QGIS
- Operações topológicas
- Processing Plugin com model builder para resolver um problema de geoprocessamento.
- (Opcional): RasterIO+Sklearn (XGBoost)

Estrutura dos projetos

- /docs
 - Diagrama de classes: Representa visualmente as principais classes que serão implementadas.
 - Fluxogramas: Representa uma visão geral de um procedimento importante.
- /src
 - Exemplos de código
- /tests
 - Testes unitários: Verificam partes do código utilizando entradas e saídas controladas.

- 6 – Implementação (entrega no classroom)
- Em pares.
- 3 – Apresentação oral do código (no computador)
- Alternado entre os dois membros do par.
- 1 – Inovação
- Ponto para ideias interessantes que o par tiver.

Cronograma de aulas

- Matemática: Numpy, Scipy
- Interface gráfica (GUI): PyQt5, Tkinter
- Análises em SIG: PyQGIS, OGR
- Dados raster geográficos: GDAL, Fiona, Rasterio,
- Dataframes (tabelas): Pandas, Geopandas, Polars
- Gráficos científicos: Matplotlib, Plotly
- Backend Web: FastAPI, Flask, Django
- Backend com interface: Taipy, Streamlit
- Aprendizado de máquina: Sklearn, Keras, Torch, Tensorflow, Jax

- Criar a pasta do projeto e acessar a pasta
- `mkdir projeto0`
- `cd projecto0`
- Inicializar o ambiente
- `pixi init`
- Ativar o ambiente criado:
- Instalando pacotes:
- `pixi add qgis`

- Prática
- Fluxograma no VSCodium
- Debug no VSCodium
- Ctrl+Shift+P
- Select Python Interpreter
- Parar debugar: F5