1.Introducao

October 27, 2019

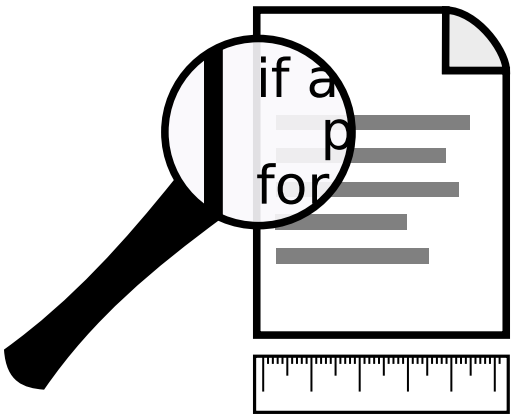
0.1 Ressalvas da versão em PDF

O minicurso foi preparado com o objetivo de ser visualizado no Jupyter, se aproveitando de scrolls em outputs e da interatividade de widgets. Em PDFs, essas funcionalidades não existem. Tentei criar uma versão alternativa para usar com PDF, mas o resultado não é o mesmo.

Se puder visualize o minicurso no formato de notebooks (disponíveis em https://github.com/JoaoFelipe/minicurso-mineracao-interativa). Você pode usar o Gitpod para executá-los.

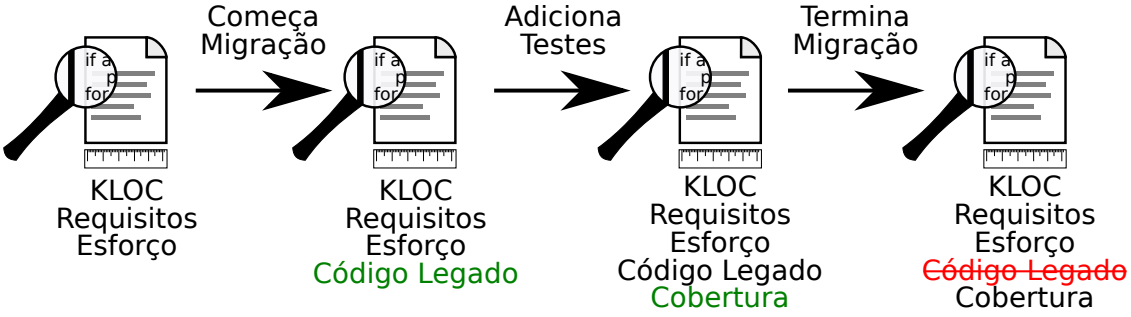
Para manter a **qualidade de software** é necessário **monitorar** e **extrair métricas**.

[22]:
```
import pdffallback
pdffallback.image("images/extractmetrics.svg")
```



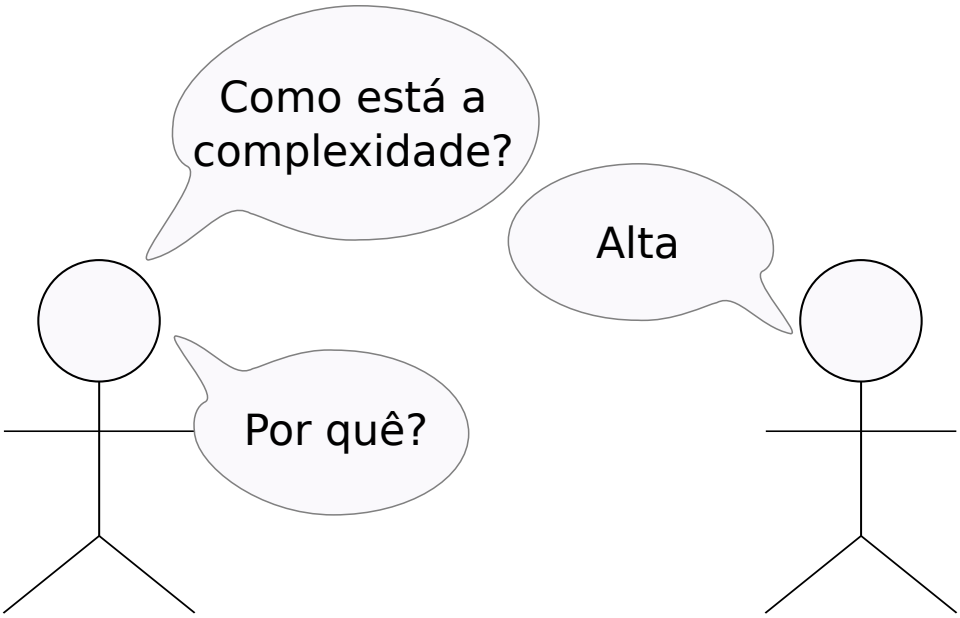
Durante a evolução do software, não só funcionalidades vão sendo adicionadas e removidas, mas o processo de desenvolvimento também muda. Às vezes novas métricas são adicionadas. Às vezes métricas antigas deixam de fazer sentido.

[23]:
```
pdffallback.image("images/evolution.svg")
```



Em algumas situações, o processo de monitoramento precisa ser **exploratório** para que se obtenham informações relevantes para o momento.

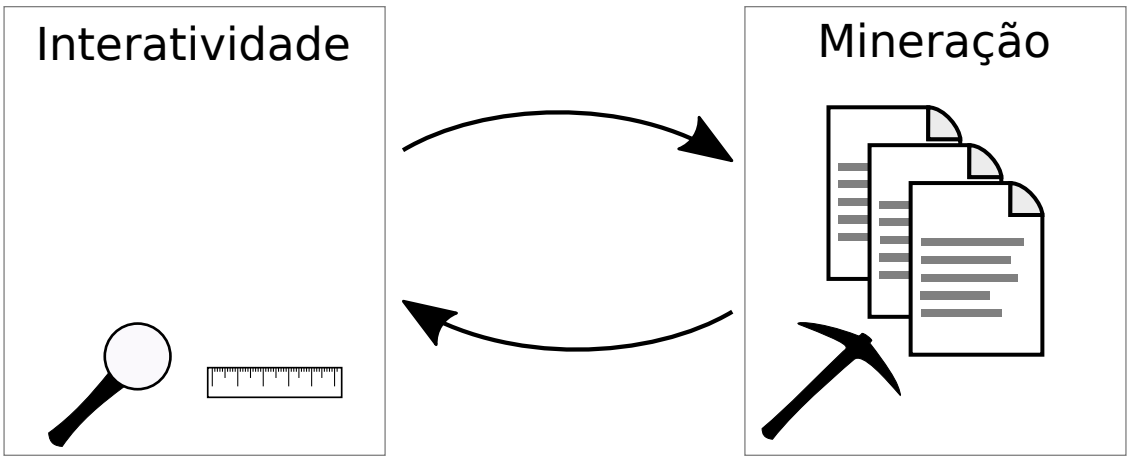
[24]:
```
pdffallback.image("images/complexity.svg")
```



Para facilitar explorações e melhorias contínuas do processo, é interessante que as análises sejam feitas de forma **interativa**, ou seja, com a possibilidade de fazer análises exploratívas e integrar análises prontas ao processo.

Para obter dados para análises, precisamos **minerar** repositórios de software sob demanda.

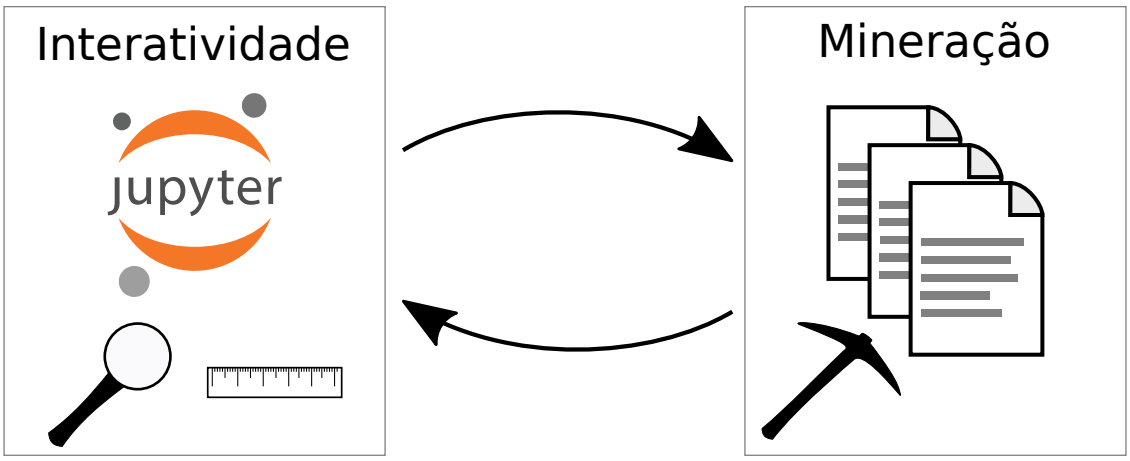
[25]:
```
pdffallback.image("images/miningv0.svg")
```



Este minicurso está dividido em 2 partes:

- Interatividade
- Mineração

[26]:
```
pdffallback.image("images/mining.svg")
```



0.2 Por que Jupyter?

Ferramenta que permite combinar código, texto, visualização, e widgets interativos.

O código fica organizado em células que podem ser executadas e re-executadas em qualquer ordem e de acordo com o desejo do usuário.

Em uma análise explorativa, é possivel manter resultados parciais, evitando esforço computacional.

Extensões ao Python facilitam algumas tarefas.

0.3 Interatividade

Será apresentado o Jupyter com as modificações ao Python proporcionadas pelo IPython, tais como **bang expressions**, **line magics** e **cell magics**.

Para visualização, usaremos a biblioteca **matplotlib** e e estenderemos a visualização rica do Jupyter para formar grafos com programa **GraphViz**.

Por fim, widgets interativos do **ipywidgets** serão apresentados.

0.4 Mineração

Para mineração, criaremos um servidor de proxy em **Flask**, usaremos a biblioteca **requests** para fazer requisições web, extrairemos informações de repositórios usando comandos do **git** e usaremos a biblioteca **Pygit2** para auxiliar a extração dessas informações.

As requisições web serão feitas com 3 objetivos:

- Obter uma página HTML e usar a biblioteca **BeautifulSoup** para extrair informações dela.
- Acessar a API v3 do GitHub, que utiliza REST

- Acessar a API v4 do GitHub, que utiliza GraphQL

0.5 Minicurso

O minicurso está disponível no GitHub:

https://github.com/JoaoFelipe/minicurso-mineracao-interativa

URL curta:

https://cutit.org/MIGHUB

Ao longo do minicurso, passarei exercícios. A melhor forma de acompanhar sem perder tempo instalando dependências é pelo GitPod

https://gitpod.io/#https://github.com/JoaoFelipe/minicurso-mineracao-interativa

URL curta:

https://cutit.org/MIGPOD

0.6 Gitpod

https://cutit.org/MIGPOD

Para entrar no GitPod, basta autorizar a conexão com uma conta do GitHub.

Ao iniciar o ambiente, digite `"echo $jupynb"` no terminal e copie e cole o resultado também no terminal:

```
jupyter notebook --NotebookApp.allow_origin=\'$(gp url 8888)\' --ip='*' --NotebookApp.token=''
```

Isso iniciará o Jupyter com toda a apresentação. Estamos no arquivo `1.Introducao.ipynb`

0.7 Agenda

Vou começar apresentando o **Jupyter** com algumas funcionalidades para interatividade, apresentadas no Notebook `2.Jupyter.pdf` e `3.IPython.pdf`.

Em seguida vou falar de **mineração de repositórios**, apresentando os notebooks `4.Proxy.pdf`, `5.Crawling.pdf`, `6.API.v3.pdf`, `7.API.v4.pdf`, `8.Git.pdf`, `9.Pygit2.pdf`.

Por fim, vou voltar para a parte de interatividade para falar sobre formas de **estender** o Jupyter e **ipywidgets**, apresentando os notebooks `10.Visualizacao.Rica.pdf` e `11.Widgets.pdf`.

Programação do dia: - **09:00 - 10:00: Minicurso** - 10:00 - 10:30: Coffee Break - **10:30 - 12:00: Minicurso** - 12:00 - 13:30: Almoço - **13:30 - 15:45: Minicurso**

0.8 Um pedido de desculpas

Descrição do minicurso

> O mini-curso tem o objetivo de apresentar mineração interativa de repositórios com o objetivo de melhoria continua de processos. O mini-curso abordará 4 tópicos: interatividade, coleta de dados, análise e visualização. Para interatividade, será apresentada a ferramenta Jupyter Notebook, indicando como ela pode ser usada para tarefas exploratórias e para a construção de dashboards. Para a coleta de dados, será usada a API do GitHub para obter issues de um repositório e a biblioteca PyGit2 para navegar no histórico. Para análise dos dados, será usada a biblioteca pandas. Por fim, para a visualização dos dados, será usada a biblioteca Matplotlib. O mini-curso será guiado por tarefas tais como observar a densidade de defeitos do projeto com o passar do tempo, descobrir quem são os desenvolvedores que mais contibuíram com o projeto no decorrer do tempo, ~~medir a cobertura de testes ao longo do tempo~~, etc.

O projeto que vou usar de exemplo ao longo da apresentação (`gems-uff/sapos`) mudou bastante ao longo do tempo e preparar um ambiente para medir **cobertura de testes** ao longo do tempo se mostrou mais complicado do que eu gostaria. Por conta disso, essa operação foi substituída por uma mais simples: medir a quantidade de linhas ao longo do tempo.

Continua: 2.Jupyter.pdf