

UNIVERSIDADE FEDERAL DE ALAGOAS  
CENTRO DE TECNOLOGIA  
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA CIVIL

WEVERTON MARQUES DA SILVA

**MODELAGEM E DESENVOLVIMENTO DE UMA APLICAÇÃO DE UMA  
APLICAÇÃO PARA ANÁLISE DUTOS EM VÃO-LIVRE**

Maceió-AL

Setembro de 2019

WEVERTON MARQUES DA SILVA

**MODELAGEM E DESENVOLVIMENTO DE UMA APLICAÇÃO DE UMA  
APLICAÇÃO PARA ANÁLISE DUTOS EM VÃO-LIVRE**

Dissertação apresentada como requisito parcial  
para obtenção do grau de Mestre pelo Programa  
de Pós-Graduação em Engenharia Civil do  
Centro de Tecnologia da Universidade Federal  
de Alagoas.

Orientador: Adeildo Soares Ramos Júnior  
Coorientador: Eduardo Setton S. da Silveira

Maceió-AL

Setembro de 2019

## **LISTA DE TABELAS**

## SUMÁRIO

<b>1</b>	<b>MODELAGEM COMPUTACIONAL . . . . .</b>	<b>4</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>7</b>

## 1 MODELAGEM COMPUTACIONAL

Neste capítulo serão apresentados os aspectos para o desenvolvimento da ferramenta, como os pré-requisitos, escolhas das ferramentas, decisões de modelagem, e as práticas adotadas no processo.

O levantamento dos requisitos de um sistema é o elemento que fornece elementos que dever nortear uma série de decisões a serem tomadas no seu desenvolvimento. A ferramenta a ser implementada deve atender a uma série de requisitos relacionados com o casos de uso a qual se propõe.

- 

softwares de cunho científico Wilson et al. (2014) apresenta um conjunto de boas práticas a serem adotadas no desenvolvimento desse tipo de. A seguir é apresentado o resumo do autor sobre essas práticas:

### **Boas práticas para computação científica**

#### **1. Escreva programas para pessoas, não para computadores.**

- a) Um programa não deve exigir que seus leitores mantenham mais de um punhado de fatos na memória de uma só vez.
- b) Torne os nomes consistentes, distintos e significativos.
- c) Tornar consistente o estilo e a formatação do código.

#### **2. Deixe o computador fazer o trabalho.**

- a) Faça o computador repetir tarefas.
- b) Salve comandos recentes em um arquivo para reutilização.
- c) Use uma ferramenta de construção para automatizar fluxos de trabalho.

#### **3. Faça alterações incrementais.**

- a) Trabalhe em pequenos passos com feedback frequente e correção de rumo.
- b) Use um sistema de controle de versão.
- c) Coloque tudo o que foi criado manualmente no controle de versão.

#### **4. Não se repita (ou repita outros).**

- a) Todos os dados devem ter uma única representação oficial no sistema.
- b) Modularize o código em vez de copiar e colar.

- c) Reutilize o código em vez de reescrevê-lo.

#### 5. Planeje erros.

- a) Adicione asserções aos programas para verificar seu funcionamento.
- b) Use uma biblioteca de testes unitários pronta para uso.
- c) Transforme erros em casos de teste.
- d) Use um depurador simbólico.

#### 6. Otimize o software somente depois que ele funcionar corretamente.

- a) Use um *profiler* para identificar gargalos.
- b) Escreva o código na linguagem de nível mais alto possível.

#### 7. Documente design e finalidade, não a mecânica.

- a) Documente interfaces e razões, não implementações.
- b) Refatore o código, em vez de explicar como ele funciona.
- c) Incorpore a documentação do software no próprio software.

#### 8. Colabore.

- a) Use revisões de código de *pré-merge*.
- b) Use a programação em pares ao interagir alguém novo e ao lidar com problemas particularmente difíceis.
- c) Use uma ferramenta de acompanhamento de problemas.

Em observância à estas práticas, algumas decisões foram tomadas quanto ao desenvolvimento da ferramenta, e serão apresentadas a seguir.

- Paradigma de programação

O uso de Programação Orientada a Objeto promove um alto índice de reaproveitamento de código, que vai da direção do item 4 apresentado.

- Linguagem de programação

A linguagem adotada será **Python**<sup>1</sup>, a qual se destaca para essa aplicação por algumas características:

- Disponibilidade de bibliotecas para aplicações científicas contemplando manipulação de matrizes (Numpy), funções matemáticas (SciPy), manipulação de dados em forma

<sup>1</sup> <https://www.python.org>

tabular (Pandas), criação de gráficos interativos (Matplotlib e Bokeh);

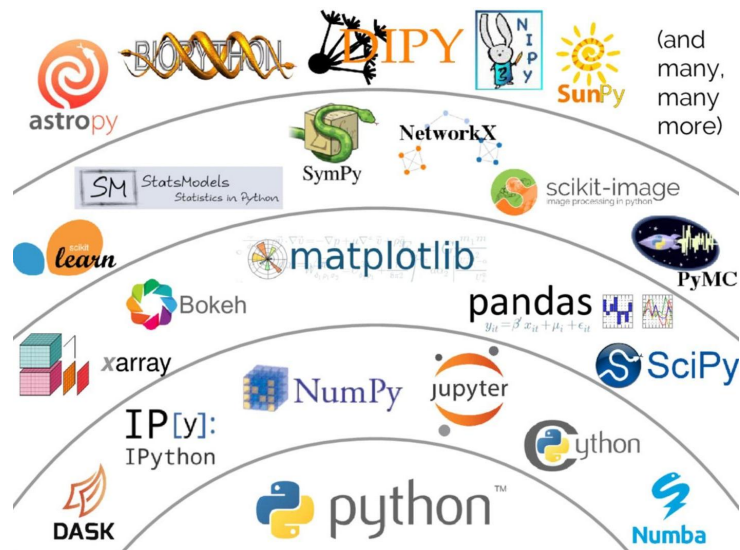


Figura 1.1 – Amostra das principais bibliotecas científicas em Python.

## REFERÊNCIAS

WILSON, G. et al. Best Practices for Scientific Computing. *PLoS Biology*, Public Library of Science, v. 12, n. 1, p. e1001745, jan 2014. ISSN 1545-7885. Disponível em: <<https://dx.plos.org/10.1371/journal.pbio.1001745>>. Citado na página 4.