



Programação de Computadores

- ✓ Apresentação do plano de ensino

Apresentação

Prof. Marco Antonio Sanches

Formação acadêmica:

- ✓ Licenciado (1995) e Pós-graduado (1997) em Matemática.
- ✓ Tecnólogo em Sistemas de Informação (2014).
- ✓ Pós-graduado em Ensino de Física (2005), Tecnologias digitais na Educação (2019) e Estatística (2002).
- ✓ Mestre em Ensino de Ciências e Matemática (2020).

Atuação profissional:

- ✓ Docente desde 1996.
- ✓ Desenvolvimento Web e Mobile (híbrido).



Apresentação



ESPAÇO DO ALUNO



Ementa

“ Estudo dos aspectos (conceitos, técnicas e ferramentas) relacionados ao desenvolvimento de programas para computador. ”



Conteúdo

Introdução:

- ✓ Apresentação e discussão do Plano de Ensino, focando objetivos, conteúdos, estratégias, avaliação e bibliografia.
- ✓ Contextualização da programação de computadores no semestre, curso, área e mercado de trabalho.

Conceitos básicos:

- ✓ Edição, compilação, execução, depuração.
- ✓ Programas e Linguagens de Programação.
- ✓ Linguagem *Python* para programação.



Conteúdo

Atribuição e E/S (Entrada e Saída):

- ✓ Comando de atribuição;
- ✓ Leitura de dados;
- ✓ Exibição de dados;
- ✓ Exemplos de algoritmos e implementação em *Python*.

Operações Numéricas:

- ✓ Tipos e operadores numéricos;
- ✓ Precedência de operadores;
- ✓ Conversão de tipos;
- ✓ Exemplos de algoritmos e implementação em *Python*.



Conteúdo

Operações lógicas:

- ✓ Tipo booleano ou lógico;
- ✓ Operações lógicas (and or not);
- ✓ Exemplos de algoritmos e implementação em *Python*.

Tomadas de decisão (estrutura condicional):

- ✓ Comando Se...então;
- ✓ Comando Se...então...Senão;
- ✓ Composição de condições (and or not);
- ✓ Condicionais aninhados;
- ✓ Exemplos de algoritmos e implementação em *Python*.



Conteúdo

Estruturas de repetição:

- ✓ Comando Enquanto...;
- ✓ Comando Faça... Enquanto;
- ✓ Comando Para... até.....;
- ✓ Composição de comandos de repetição e condicionais;
- ✓ Comandos de repetição aninhados;
- ✓ Exemplos de algoritmos e implementação em Python.

Processamento de textos (operações com cadeias de caracteres):

- ✓ Conceito de String;
- ✓ Operações com Strings;
- ✓ Exemplos de algoritmos e implementação em Python.



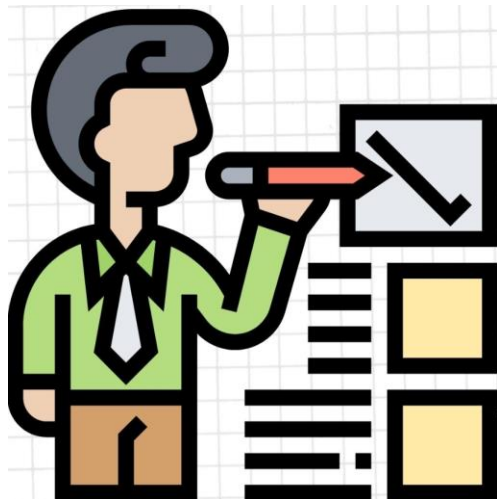
Conteúdo

Vetores - estruturas lineares:

- ✓ Conceito de vetor;
- ✓ Operações básicas em vetores (criação inicialização inserção remoção troca e busca);
- ✓ Exemplos de algoritmos e implementação em Python.



Avaliação e horário de aulas

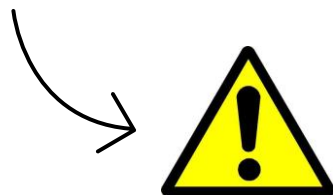


- ✓ Avaliação Regimental - 5,0 →
- ✓ Avaliação Parcial - 3,0
- ✓ Exercícios/Projetos - 2,0

A1

A2

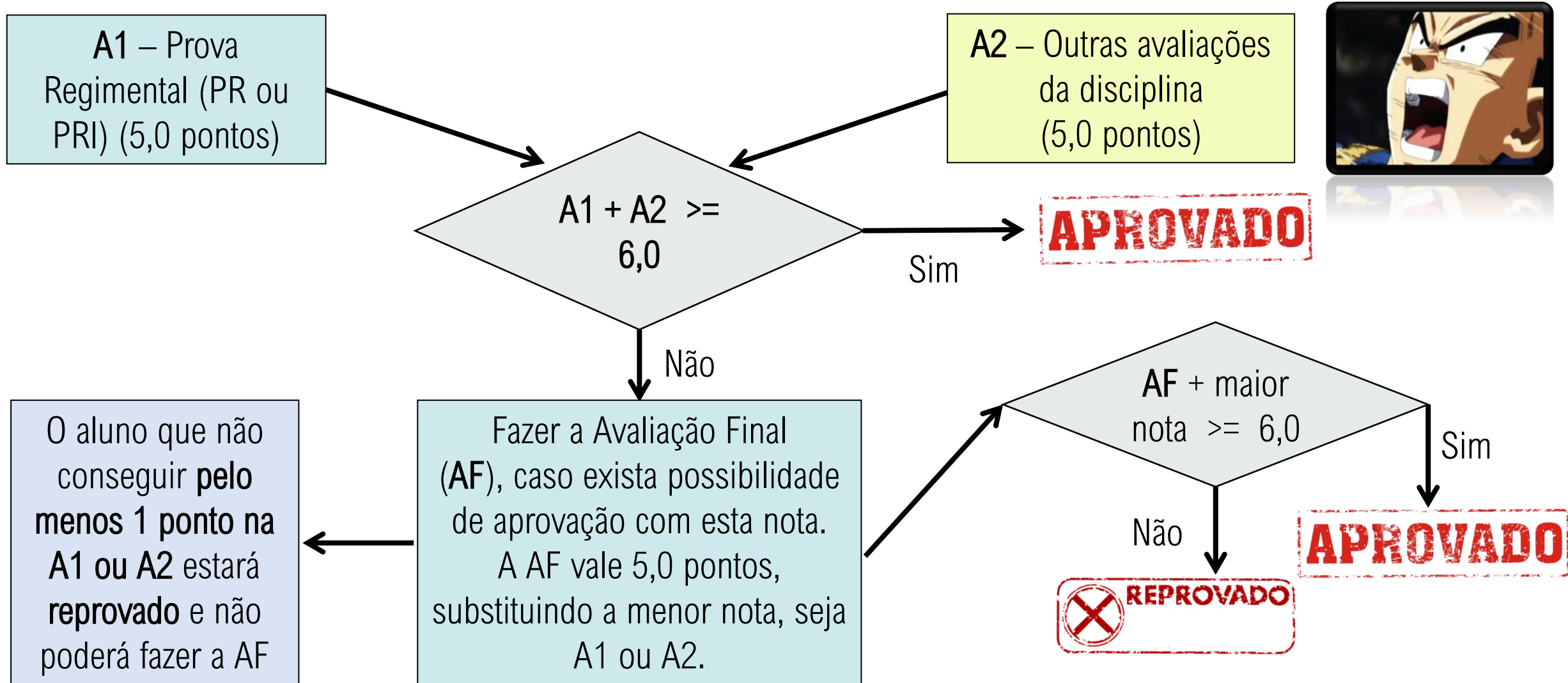
Nota Final = A1 + A2



HORÁRIOS DE AULA

- ✓ Manhã: 08:30 às 9:45 / 9:55 às 11:10
- ✓ Noite: 19:10 às 20:25 / 20:35 às 21:50


Sistema de avaliação



Bibliografia

Básica:

Bibliografia Básica



1) FORBELLONE, A. L. V.; EBERSPACHER, H. F. **Lógica de Programação: A Construção de Algoritmos e Estrutura de Dados**. 3. ed. São Paulo: Pearson Prentice Hall, 2008.

Prog. Computadores – 1º semestre 2023

13

Bibliografia Básica




2) MANZANO, José Augusto N. G. **Algoritmos: lógica para desenvolvimento de programação de computadores**. 28. ed. São Paulo: Erica, 2016 (e-book).

Prog. Computadores – 1º semestre 2023

14

Bibliografia Básica



3) VILARIM, G. O. **Algoritmos: Programação Para Iniciantes**. Rio de Janeiro: Ciência Moderna, 2004.

Prog. Computadores – 1º semestre 2023

15

Complementar:

Bibliografia Complementar

ASCENCIO, Ana F. G.; CAMPOS, Edilene A. V. **Fundamentos da Programação de Computadores** - 3ª edição São Paulo: Pearson, 2012. (e-book).

DASGUPTA, SANJOY; PAPADIMITRIOU, CHRISTOS; VAZIRANI, UMESH. **Algoritmos**. Porto Alegre: Grupo A, 2011. (e-book).

PYTHON SOFTWARE FOUNDATION. **The Python Language Reference**. Disponível em: <https://docs.python.org/3/reference/index.html> Acesso em: 12 nov. 2018.

PYTHON SOFTWARE FOUNDATION. **The Python Standard Library**. Disponível em: <https://docs.python.org/3/library/index.html> Acesso em: 12 nov. 2018.

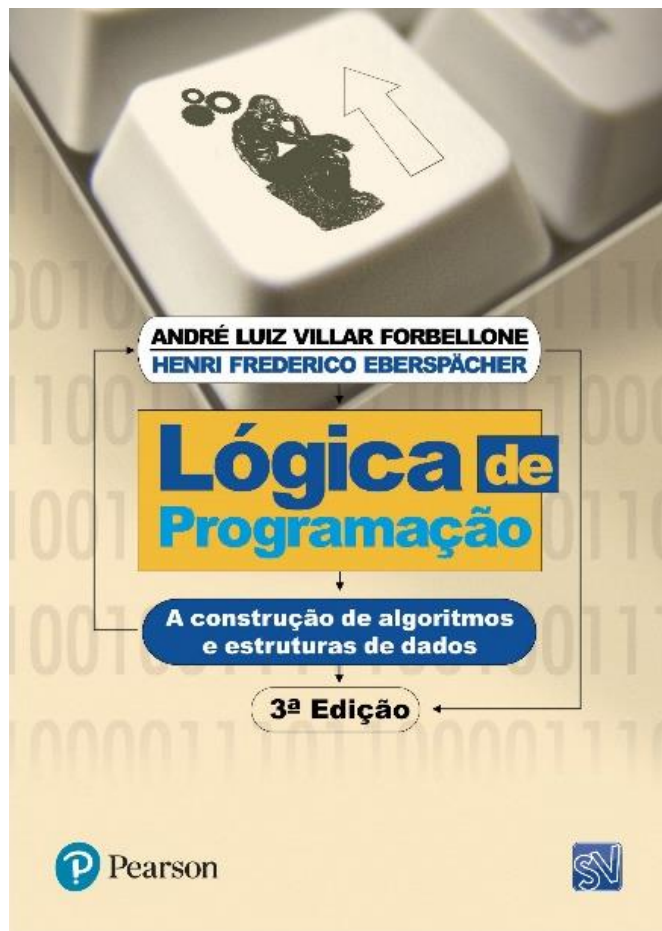
PYTHON SOFTWARE FOUNDATION. **The Python Tutorial**. Disponível em: <https://docs.python.org/3/tutorial/index.html> Acesso em: 12 nov. 2018.

+ Materiais de aula (resumo).
+ Materiais disponíveis na Internet e indicados pelos professores quando for pertinente.

Prog. Computadores – 1º semestre 2023

16

Bibliografia Básica



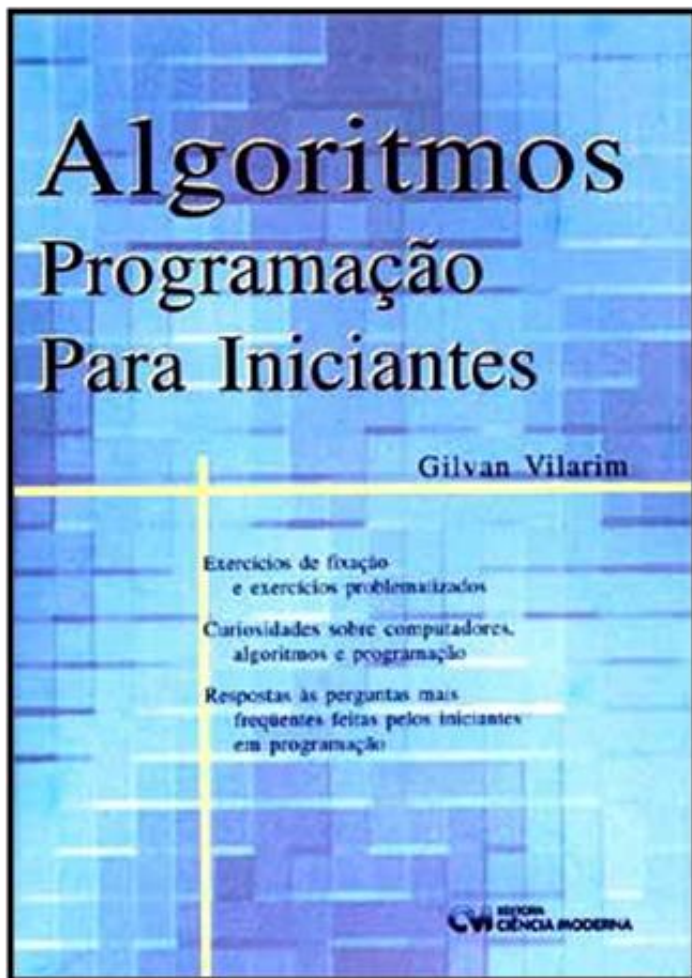
1) FORBELLONE, A. L. V.; EBERSPACHER, H. F. **Logica de Programação: A Construção de Algoritmos e Estrutura de Dados.** 3. ed. São Paulo: Pearson Prentice Hall, 2008.

Bibliografia Básica



2) MANZANO, José Augusto N. G. Algoritmos: lógica para desenvolvimento de programação de computadores. 28. ed. São Paulo: Erica, 2016 (e-book).

Bibliografia Básica



3)VILARIM, G. O. **Algoritmos: Programação Para Iniciantes**. Rio de Janeiro: Ciência Moderna, 2004.

Bibliografia Complementar

ASCENCIO, Ana F. G.; CAMPOS, Edilene A. V. Fundamentos da Programação de Computadores - 3ª edição São Paulo: Pearson, 2012. (e-book).

DASGUPTA, SANJOY; PAPADIMITRIOU, CHRISTOS; VAZIRANI, UMESH. Algoritmos. Porto Alegre: Grupo A, 2011. (e-book).

PYTHON SOFTWARE FOUNDATION. The Python Language Reference. Disponível em: <https://docs.python.org/3/reference/index.html> Acesso em: 12 nov. 2018. PYTHON SOFTWARE

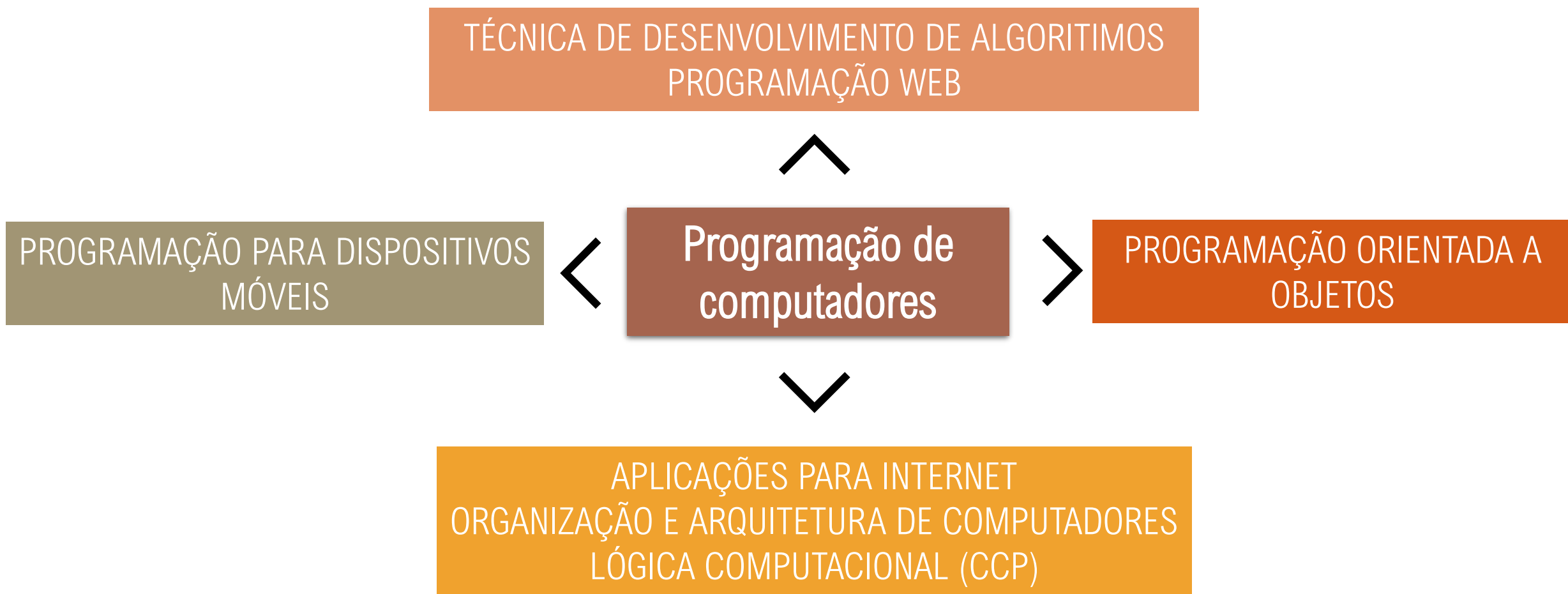
FOUNDATION. The Python Standard Library. Disponível em: <https://docs.python.org/3/library/index.html>. Acesso em: 12 nov. 2018. PYTHON SOFTWARE

FOUNDATION. The Python Tutorial. Disponível em: <https://docs.python.org/3/tutorial/index.html> Acesso em: 12 nov. 2018.

+ Materiais de aula (resumo).

+ Materiais disponíveis na Internet e indicados pelos professores quando for pertinente.

Relação (diretas) com disciplinas do curso



Linguagem de programação Python

Nesta disciplina iremos utilizar a linguagem de programação **Python**.



A linguagem Python foi concebida no final de 1989 no Instituto de Pesquisa Nacional para Matemática e Ciência da Computação em Amsterdã.



** https://pt.wikipedia.org/wiki/Guido_van_Rossum

Ferramentas

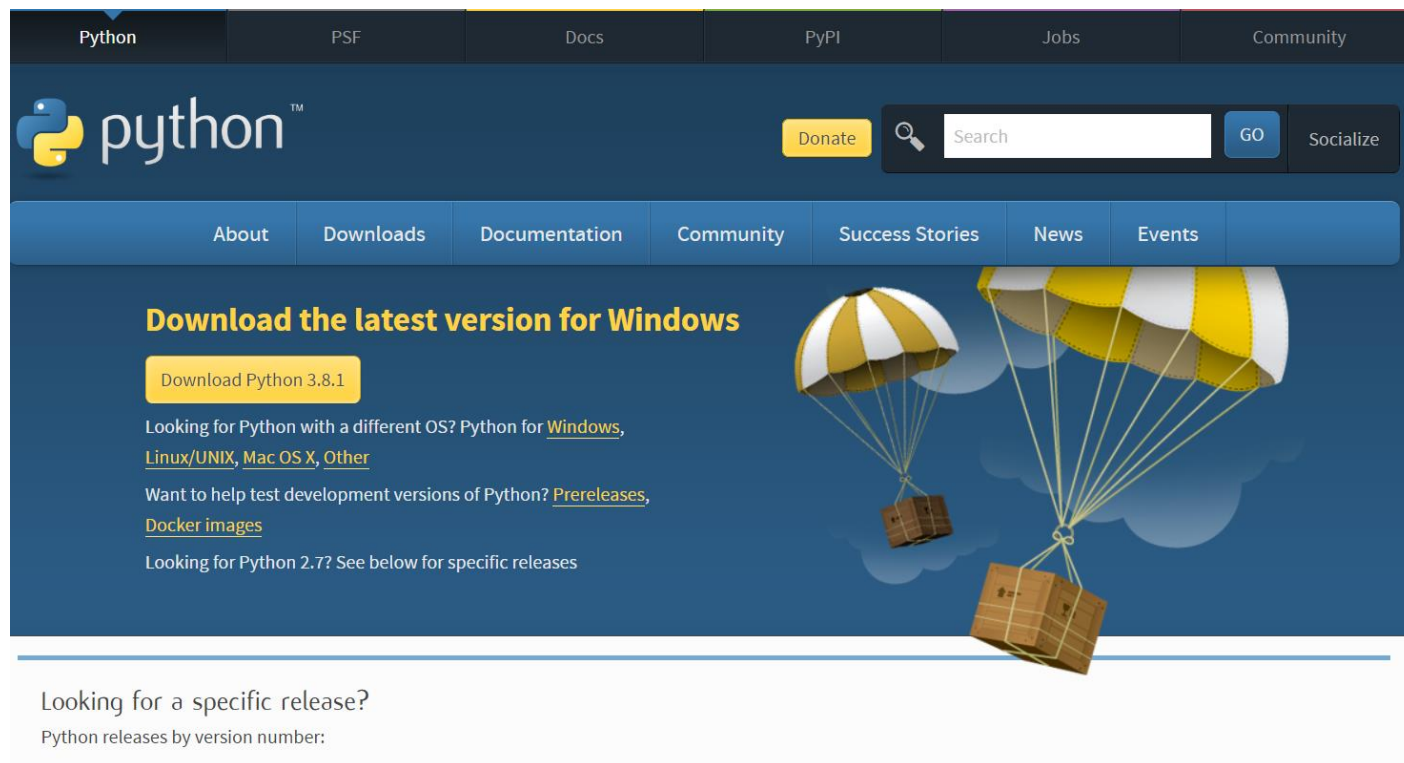
Nesta disciplina iremos utilizar as seguintes ferramentas:

- ✓ Laboratório
- ✓ Caderno/Lápis/Borracha
- ✓ Python



Ferramentas – Python

Acesse o site do Python para baixar e instalar a versão 3.10 ou superior:

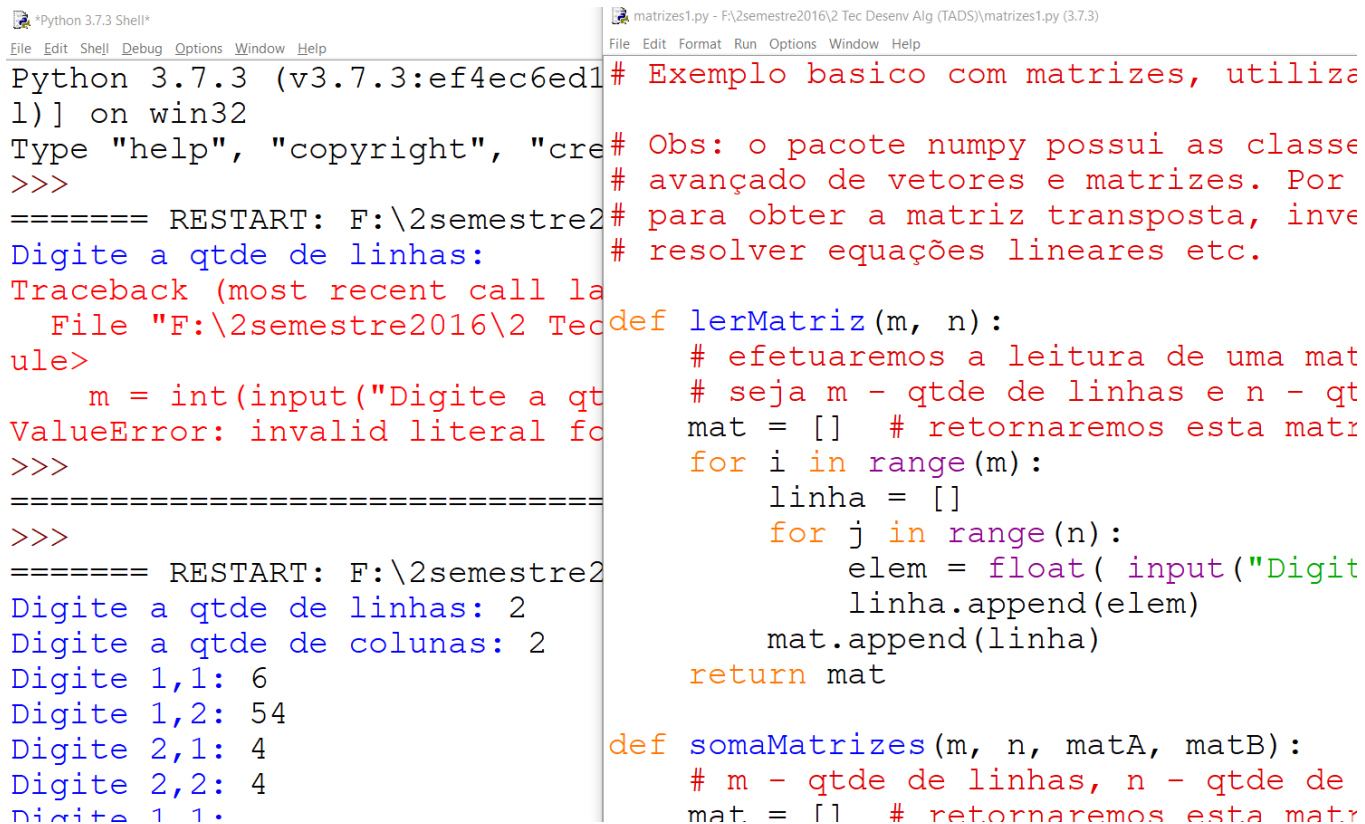


<https://www.python.org/downloads/>



Ferramentas – Python

Após instalar o Python, teremos a opção de usar a linha de comando ou o IDLE*



```
*Python 3.7.3 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed1) on win32
Type "help", "copyright", "credits()" or "quit()" for more
>>>
===== RESTART: F:\2semestre2016\2 Tec Desenv Alg (TADS)\matrizes1.py (3.7.3)
Digite a qtde de linhas:
Traceback (most recent call last):
  File "F:\2semestre2016\2 Tec Desenv Alg (TADS)\matrizes1.py", line 1, in <module>
    m = int(input("Digite a qtde de linhas: "))
ValueError: invalid literal for int() with base 10: '54'
>>>
===== RESTART: F:\2semestre2016\2 Tec Desenv Alg (TADS)\matrizes1.py (3.7.3)
Digite a qtde de linhas: 2
Digite a qtde de colunas: 2
Digite 1,1: 6
Digite 1,2: 54
Digite 2,1: 4
Digite 2,2: 4
Digite 1 1.

matrizes1.py - F:\2semestre2016\2 Tec Desenv Alg (TADS)\matrizes1.py (3.7.3)
File Edit Format Run Options Window Help
# Exemplo basico com matrizes, utiliza
# Obs: o pacote numpy possui as classe
# avançado de vetores e matrizes. Por
# para obter a matriz transposta, inve
# resolver equações lineares etc.

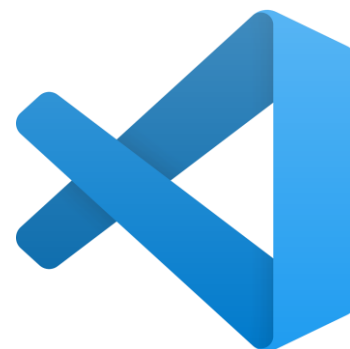
def lerMatriz(m, n):
    # efetuaremos a leitura de uma mat
    # seja m - qtde de linhas e n - qt
    mat = [] # retornaremos esta matr
    for i in range(m):
        linha = []
        for j in range(n):
            elem = float(input("Digit
            linha.append(elem)
        mat.append(linha)
    return mat

def somaMatrizes(m, n, matA, matB):
    # m - qtde de linhas, n - qtde de
    mat = [] # retornaremos esta matr
```

* IDLE é um ambiente de desenvolvimento integrado para Python

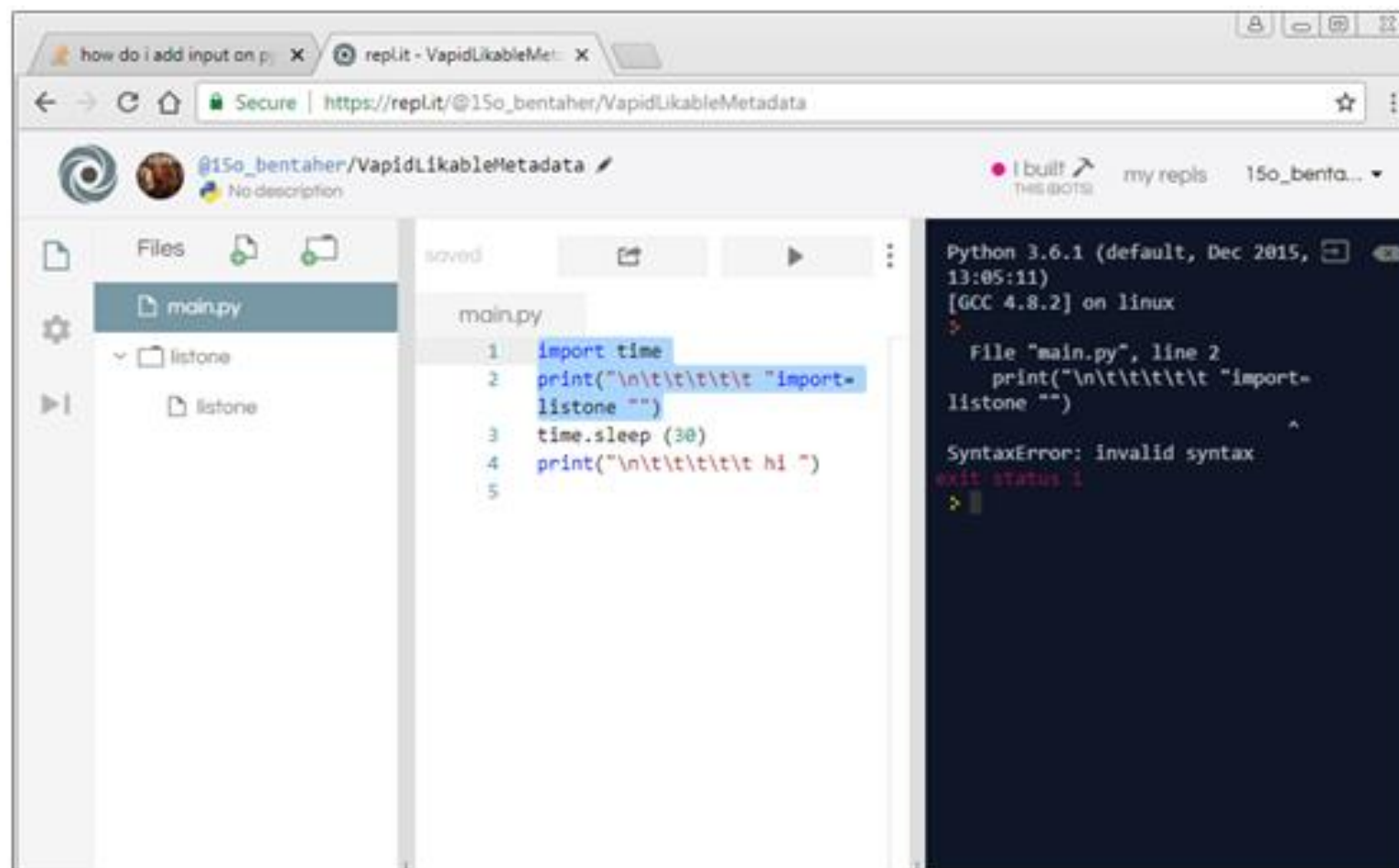
Python IDLE / Editor de código

1. PyCharm
2. Spyder
3. Sublime
4. Jupyter
5. Atom
6. Python IDLE
7. VSCode
8. Colaboratory



Opções On-line

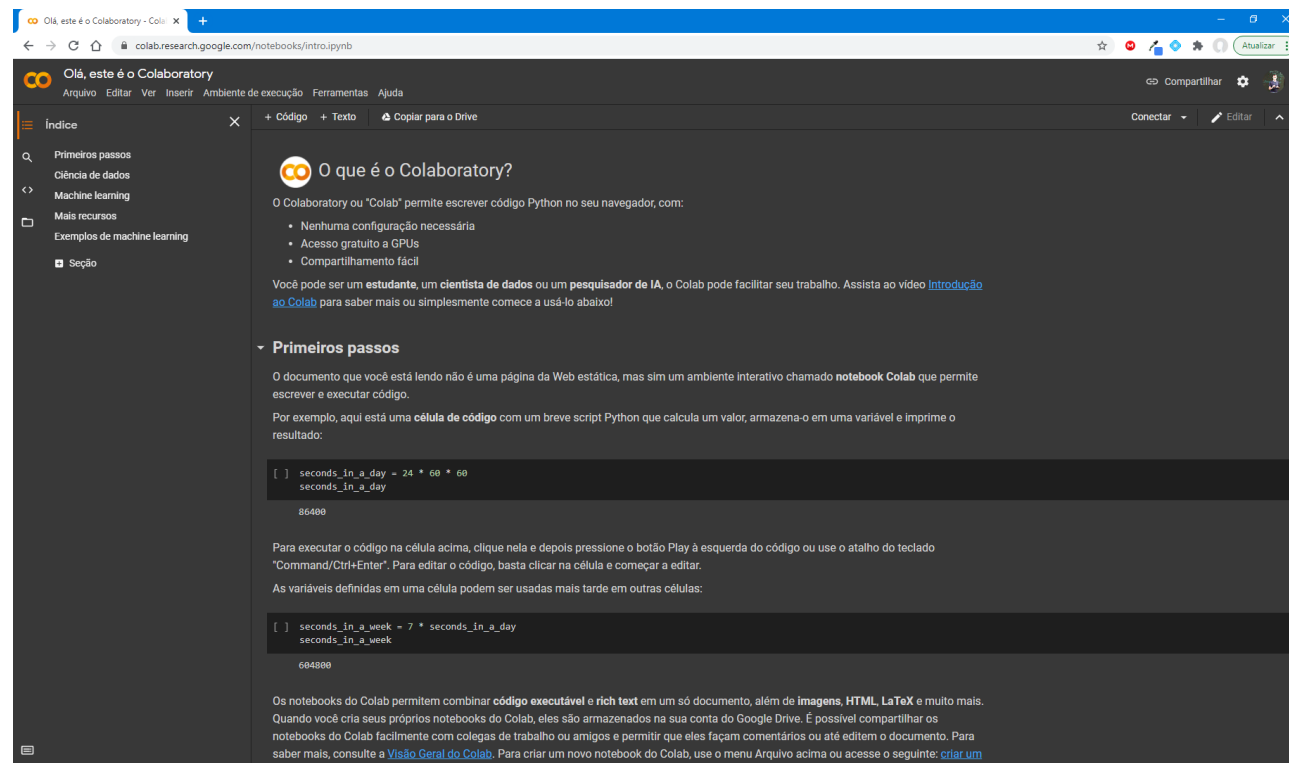
<https://repl.it/languages/python3>



Opções On-line

O Colaboratory ou "Colab" permite escrever código Python no seu navegador, com:

- ✓ Nenhuma configuração necessária
- ✓ Acesso gratuito a GPUs
- ✓ Compartilhamento fácil

The logo for Google Colaboratory, featuring the word "colab" in a bold, orange, sans-serif font. The letters are slightly rounded and have a subtle gradient.

<https://colab.research.google.com/>

Outras opções



<https://www.programiz.com/python-programming/online-compiler/>



<https://www.w3schools.com/python/default.asp>



<https://docs.python.org/3/reference/index.html>

ferramentas



Exemplos e exercícios resolvidos no GitHub

<https://github.com/msanches/ProgComp>

O Que é GitHub e Para Que é Usado?

<https://www.hostinger.com.br/tutoriais/o-que-github/>

Tutorial do GIT Básico – Introdução ao GIT

<https://www.hostinger.com.br/tutoriais/tutorial-do-git-basics-introducao/>



Hello World – do GitHub

<https://guides.github.com/activities/hello-world/>



GitHub Training & Guides

<https://www.youtube.com/githubguides>



Git e Github para iniciantes

<https://www.udemy.com/course/git-e-github-para-iniciantes/>

Por que todos deveriam aprender a programar?

<https://www.youtube.com/watch?v=mHW1Hsqlp6A>



Por que todos deveriam aprender a programar?

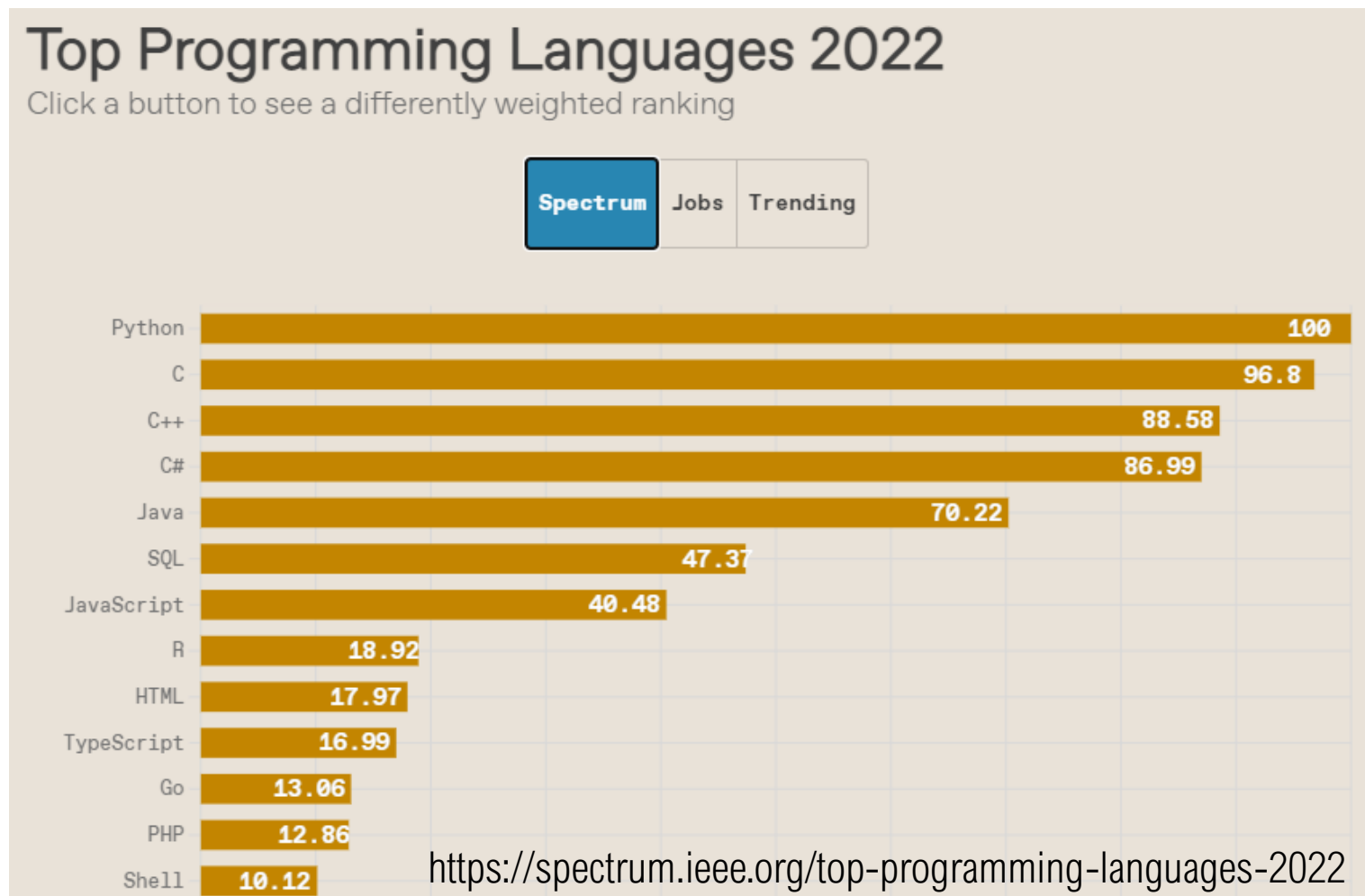


Linguagens de programação?!













Linguagens de programação?!

IEEE Spectrum's Top Programming Languages 2022



Linguagens de programação?!

Dec 2022	Dec 2021	Change	Programming Language		Ratings	Change
1	1			Python	16.66%	+3.76%
2	2			C	16.56%	+4.77%
3	4	^		C++	11.94%	+4.21%
4	3	v		Java	11.82%	+1.70%
5	5			C#	4.92%	-1.48%
6	6			Visual Basic	3.94%	-1.46%
7	7			JavaScript	3.19%	+0.90%
8	9	^		SQL	2.22%	+0.43%
9	8	v		Assembly language	1.87%	-0.38%
10	12	^		PHP	1.62%	+0.12%



<https://www.tiobe.com/tiobe-index/>

Desafios? Aprender mais?

BEECROWD



Somos uma comunidade global de desenvolvedores comprometidos em continuar evoluindo como estudantes e profissionais. Treine algoritmos e desafios de programação e torne-se o especialista que você sempre sonhou ser.

PÁGINA CORPORATIVA



beecrowd

USAR LOGIN SOCIAL



COMPETIÇÕES E RANKING

Junte-se às mentes mais brilhantes da programação competitiva! Participe de competições, concursos e torneios! Compare seus conhecimentos com os de seus colegas. Suba de nível, cresça e brilhe em sua carreira!



<https://www.beecrowd.com.br/judge/pt/login>

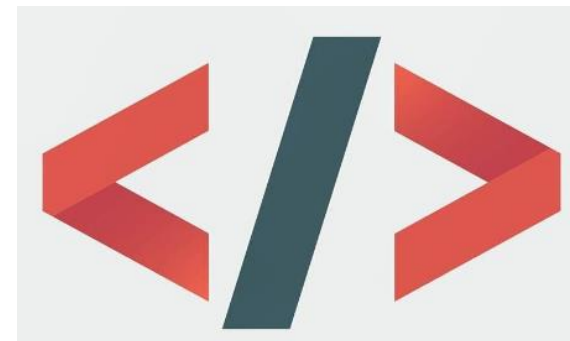
Resolva os problemas disponíveis utilizando as 11 linguagens de programação, competindo com os outros usuários. Como desafio, melhore seu ranking, resolvendo o máximo de problemas e aperfeiçoando seu código fonte.

Reflexão (motivação)



12 dicas para um Programador iniciante

1. Tenha calma, respeite as fases de aprendizado.
2. Dê valor aos conceitos.
3. Faça parte de comunidades de programadores
4. Pergunte sempre que tiver dúvidas.
5. Foque em uma linguagem / tecnologia de cada vez.
6. Não pare o estudo de uma linguagem no meio para “pular” para outra.
7. Divida a aprendizagem entre estudo e prática.
8. Seja o melhor programador júnior que puder ser.
9. Crie o seu portfólio já na fase dos estudos (utilize o GitHub e LinkedIn).
10. Compartilhe seus conhecimentos.
11. Digite o código... não copie e cole.
12. Tenha metas e recomece sempre que for preciso.



Alguma dúvida????





Vamos
começar???
