

UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE MATEMÁTICA - IM
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA - PPGMAT
PLANO DE CURSO

CURSO: Pensamento Computacional com Python

CARGA HORÁRIA SEMESTRAL: 42h

CARGA HORÁRIA SEMANAL: 3h

COORDENAÇÃO: Alexandre de Sousa

HORÁRIO: *a definir*

MOTIVAÇÃO:

Pensamento Computacional é o processo de pensamento envolvido na formulação de um problema e na expressão de sua solução de forma que um computador (humano ou máquina) possa efetivamente realizar. Ocorre que aprender a programar é uma excelente oportunidade de praticar e desenvolver pensamento computacional. Em vez de usar programas prontos como ferramentas auxiliares em nossas tarefas, nós tratamos *computadores como máquinas a serem ensinadas*, e usamos linguagens de programação como uma ferramenta para expressar o que desejamos ensinar. O ato de programar combina algumas das melhores características da matemática, das engenharias e das ciências naturais; como na matemática, se usa linguagens formais para representar ideias (especificamente, computações); como nas engenharias, se projeta coisas, montando sistemas a partir de componentes e avaliando as vantagens e desvantagens entre as alternativas; e, como nas ciências naturais, se observa o comportamento de sistemas complexos, se formula hipóteses e testa previsões. Programação exige sermos precisos e não ambíguos, sem ser excessivamente rigorosos. Dessa forma, o ato de programar torna-nos conscientes de nossos erros de raciocínio ou conclusões infundadas, encorajando o pensamento claro na formulação, expressão e execução de ideias na forma computacional.

OBJETIVOS:

Colocar as pessoas participantes em contato com o processo de pensamento envolvido no Pensamento Computacional, usando a linguagem Python como ferramenta pedagógica. Em um certo nível, as pessoas aprenderão a programar, uma habilidade que é útil em si mesma; em outro nível, as pessoas desenvolverão as habilidades de formular questões, pensar criativamente em soluções possíveis e expressar essa solução de forma clara e precisa. Ao final deste curso, as pessoas participantes deverão ser capazes de:

- Compreender a terminologia relacionada a linguagens de programação;
- Compreender os conceitos e as técnicas fundamentais usualmente aplicadas na expressão de ideias computacionalmente;
- Expressar-se na linguagem de programação Python;
- Selecionar as ferramentas adequadas para abordar questões de seu interesse usando uma linguagem de programação;
- Usar linguagens de programação como instrumento de aprendizagem de ideias e conceitos que possam ser expressos computacionalmente (física, matemática, linguística, estatística, ciência de dados, inteligência artificial etc).

PROGRAMA

Semana	Carga Horária	Assunto
1.	3	Introdução aos conceitos de linguagens de programação e da metodologia (construcionismo) adotada no curso
2.	3	Apresentação da linguagem Python
3.	3	Depuração
4.	3	Funções
5.	3	Estruturas de decisão
6.	3	Estruturas de repetição
7.	3	Tipos de dados: cadeias de caracteres
8.	3	Tipos de dados: listas e tuplas
9.	3	Arquivos: busca, leitura, escrita
10.	3	Tipos de dados: dicionários
11.	3	Recursão
12.	3	Programação orientada a objetos
13.	3	Programação funcional
14.	3	Apresentações de projetos

BIBLIOGRAFIA:

Brad Miller, e David Ranum. *Como Pensar Como um Cientista da Computação: Edição Interativa em Python*. <https://panda.ime.usp.br/pensepy/static/pensepy/index.html>. Acessado 04 de março de 2020.

Brad Miller, e Paul Resnick. *Foundations of Python Programming*. <https://runestone.academy/runestone/books/published/fopp/index.html>. Acessado 4 de março de 2020.

Wing, Jeannette M. "Computational thinking". *Communications of the ACM*, vol. 49, nº 3, março de 2006, p. 33–35. *ACM Digital Library*, doi:[10.1145/1118178.1118215](https://doi.org/10.1145/1118178.1118215).

Sussman, Gerald Jay, et al. *Functional differential geometry*. The MIT Press, 2013. <https://mitpress.mit.edu/books/functional-differential-geometry>.

Sussman, Gerald Jay, e Jack Wisdom. *Structure and Interpretation of Classical Mechanics*. 2. ed, MIT Press, 2014. *Gemeinsamer Bibliotheksverbund ISBN*, <https://mitpress.mit.edu/books/structure-and-interpretation-classical-mechanics-second-edition>.

Abelson, Harold, et al. *Structure and Interpretation of Computer Programs*. 2. ed, MIT Press [u.a.], 1996. <https://mitpress.mit.edu/books/structure-and-interpretation-computer-programs-second-edition>.

Zimmermann, Paul, et al. *Computational Mathematics with SageMath*. SIAM, 2018. *hal.inria.fr*, doi:[10.1137/1.9781611975468](https://doi.org/10.1137/1.9781611975468).