



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA BAHIA
ROD BR 367 KM 57,5 - Bairro Fontana I - CEP 45810-000 - Porto Seguro - BA - www.portal.ifba.edu.br

CURSO: Técnico em Informática

DOCENTE: Diêgo Braga Monteiro de Moura

DISCIPLINA: Lógica de Programação I / 2026

EMENTA

Estudo dos fundamentos da lógica de programação com ênfase no desenvolvimento do raciocínio lógico e na resolução estruturada de problemas. Interpretação e análise de situações-problema, incluindo desafios inspirados na Olimpíada Brasileira de Informática (OBI). Construção e representação de algoritmos por meio de descrição narrativa, pseudocódigo e fluxogramas. Introdução à implementação computacional utilizando a linguagem Python, abordando: sintaxe básica; execução de programas; entrada e saída de dados; variáveis e tipos de dados primitivos (inteiro, real, booleano e string); operadores aritméticos, relacionais e lógicos; expressões; estruturas condicionais (if, elif, else); estruturas de repetição (while e for); listas e operações básicas com listas; dicionários e operações básicas com dicionários; funções (definição, parâmetros e retorno); noções de modularização; e utilização introdutória de bibliotecas padrão.

OBJETIVOS

Desenvolver no estudante o raciocínio lógico e a capacidade de resolução estruturada de problemas por meio da análise, construção e representação de algoritmos, bem como da implementação dessas soluções na linguagem Python, utilizando conceitos fundamentais como variáveis, tipos de dados, operadores, estruturas de controle, funções e estruturas de dados básicas.

CONTEÚDO PROGRAMÁTICO

1. Introdução Básica
 - a. Conceito de Pensamento Computacional
 - b. Conceito de Algoritmo
 - c. Formas de Representação de um Algoritmo
 - d. Resolução de Problemas de Raciocínio Lógico
2. Fluxogramas
3. Introdução ao Python
 - a. O que é python ? Quais as características da linguagem ?
 - b. Instalação do Python + IDE
4. Tipos de Dados e Variáveis
5. Entrada e Saída
6. Operadores
 - a. Atribuição
 - b. Aritméticos
 - c. Relacionais
 - d. Lógicos
7. Estruturas Condicionais
8. Estruturas de Repetição

- 9. Funções
- 10. Bibliotecas
- 11. Estruturas de Dados (Listas/Dicionários)

METODOLOGIA

A disciplina será desenvolvida por meio de aulas expositivas dialogadas, com apresentação dos fundamentos teóricos da lógica de programação, seguidas predominantemente por atividades práticas voltadas à resolução de exercícios e situações-problema.

As atividades práticas terão como foco a interpretação de enunciados, a construção de algoritmos e a posterior implementação em linguagem Python. Será incentivada a participação ativa dos estudantes na análise e discussão das soluções propostas.

Haverá também o desenvolvimento de projeto prático ao longo do período letivo, no qual os estudantes deverão implementar a solução de um problema proposto. Durante essa etapa, o professor realizará acompanhamento contínuo, promovendo discussões orientadas sobre o andamento do projeto, esclarecendo dúvidas, analisando estratégias adotadas e estimulando a reflexão sobre as decisões algorítmicas tomadas.

SISTEMA DE AVALIAÇÃO

A avaliação da aprendizagem será realizada por meio de três instrumentos distintos, conforme normativas institucionais, contemplando diferentes dimensões do processo formativo:

I – Projeto prático, desenvolvido em sala de aula, envolvendo a implementação de uma solução para um problema real implementado em Python. A atribuição de nota estará condicionada à demonstração de domínio dos conceitos e da lógica empregada na solução apresentada, por meio de verificação escrita e/ou oral. O funcionamento do programa será considerado como parte do processo avaliativo, não sendo a simples entrega do código critério suficiente para pontuação.

II – Atividade prática de programação em laboratório, consistindo na resolução de problema computacional com possibilidade de testes a partir de entradas e saídas fornecidas, visando à validação da solução desenvolvida.

III – Prova teórica escrita, abordando conceitos fundamentais da lógica de programação, análise de algoritmos e estruturas computacionais estudadas.

RECUPERAÇÃO DAS APRENDIZAGENS

A recuperação da aprendizagem será ofertada ao longo do período letivo, com caráter formativo, visando à consolidação dos conteúdos e ao desenvolvimento das competências previstas na disciplina.

Será assegurada ao estudante a oportunidade de recuperação referente a um dos seguintes instrumentos avaliativos: atividade prática de laboratório ou prova teórica escrita, considerando aquele em que obteve menor desempenho. Nos casos em que o estudante obteve notas iguais nesses dois instrumentos, poderá optar por qual deles deseja realizar a recuperação.

A atividade de recuperação manterá a natureza do instrumento originalmente aplicado (teórica ou prática) e abrangerá os conteúdos e habilidades correspondentes.

A nota obtida na recuperação substituirá a nota anterior apenas se for superior à

BIBLIOGRAFIA

Referências Básicas

BORGES, Luiz Eduardo. *Python para Desenvolvedores*. São Paulo: Novatec, 2010.

MENEZES, Nilo Ney Coutinho. *Introdução à Programação com Python: Algoritmos e Lógica de Programação Para Iniciantes*. São Paulo: Novatec, 2014.

LUCK, Mark Lutz & ASCHER, David. *Aprendendo Python* (PDF).
https://drive.google.com/file/d/1y0QbfM4vYLbvxyum6IMG1N6POFVzb5jR/view?usp=drive_link

Tutorial de Flowcharts – Lucidchart (conceito e símbolos)
<https://www.lucidchart.com/pages/pt/o-que-e-um-fluxograma>
(explica o que é fluxograma, quando e como usar, com exemplos de símbolos e estrutura)

OLIMPIÁDA BRASILEIRA DE INFORMÁTICA (OBI) – site oficial:
<https://olimpiada.ic.unicamp.br/>

Referências Complementares

RAMALHO, Luciano. *Python Fluente: Programação Clara, Concisa e Eficaz*. São Paulo: Novatec, 2015.

ZELLIJ, John V. Guttag. *Introduction to Computation and Programming Using Python*. MIT Press, 2013. (opcional, para aprofundamento)

GADDIS, Tony. *Starting Out with Python*. Pearson, 2015. (material complementar de lógica e estruturas básicas)

EPPE, Marco & SCHIMMEL, K. D. *Fluxograma, Pseudocódigo e Linguagem C: Algoritmos e Programação*. Ciência Moderna, 2012. (bom para aprofundar representação de algoritmos, mesmo que não Python)