



Plano de Ensino

1 Código e nome da disciplina

DGT0286 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Carga horária semestral

3 Carga horária semanal

4 Perfil docente

O docente deve ser graduado em Ciência da Computação, Informática, Engenharias, com forte aderência em programação de computadores.

O Professor deve possuir Pós-Graduação Lato Sensu, sendo desejável Pós-Graduação Stricto Sensu (Mestrado ou Doutorado) na área do curso ou afins.

É desejável que o professor possua experiência de na docência no nível superior da disciplina de pelo menos 3 anos e que também tenha experiência profissional no mercado de trabalho de programação de computadores. Deve possuir familiaridade com as ferramentas digitais que fazem parte do modelo de ensino da instituição (SGC, SIA, SAVA, BDQ) além de conhecer o Projeto Pedagógico dos Cursos dos quais a disciplina faz parte na Matriz Curricular.

É necessário que o docente conheça e aplique as metodologias ativas inerentes à educação por competências e ferramentas digitais que tornam a sala de aula mais interativa e interessante para o aluno. A articulação entre a teoria e prática deve ser o eixo direcionador das estratégias em sala de aula. Além disto, é indispensável que o docente estimule o autoconhecimento e autoaprendizagem entre seus alunos.

5 Ementa

PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO: MOTIVAÇÃO E PRELIMINARES. NOMES, VINCULAÇÕES E ESCOPO. TIPOS DE DADOS. EXPRESSÕES E SENTENÇAS DE ATRIBUIÇÃO. SUBPROGRAMAS. PARADIGMAS: ESTRUTURADO, ORIENTADO A OBJETOS, FUNCIONAL E LÓGICO.

6 Objetivos

- Distinguir as categorias de linguagens de programação, fazendo a reflexão sobre os projetos de linguagens de programação, suas categorias e métodos de implementação, para decidir qual paradigma de linguagem de programação utilizar, conforme a classe de problema;

- Caracterizar a natureza dos nomes e palavras especiais nas linguagens de programação, baseando-se na linguagem Python, para empregar as regras de escopo;
- Especificar variáveis, empregando tipos de dados, de forma a contextualizar ao compilador/interpretador como o programador pretende utilizar os dados;
- Empregar formas fundamentais de instruções, baseando-se na sintaxe e semântica de expressões aritméticas, relacionais e booleanas e atribuições, para escrever instruções matemática e lógicas compreensíveis, corretas e executáveis por computadores;
- Escrever programas modularizados, baseando-se em fundamentos de subprogramas, para decompor problemas complexos em fragmentos mais simples, ou seja, mais facilmente tratáveis, cujos códigos sejam reutilizáveis e manuteníveis;
- Praticar a codificação de soluções, utilizando diferentes paradigmas de linguagem de programação, para resolver problemas aplicando o paradigma mais apropriado.

7 Procedimentos de ensino-aprendizagem

Aulas interativas em ambiente virtual de aprendizagem, didaticamente planejadas para o desenvolvimento de competências, tornando o processo de aprendizado mais significativo para os alunos. Na sala de aula virtual, a metodologia de ensino contempla diversas estratégias capazes de alcançar os objetivos da disciplina.

Os temas das aulas são discutidos e apresentados em diversos formatos como leitura de textos, vídeos, hipertextos, links orientados para pesquisa, estudos de caso, podcasts, atividades animadas de aplicação do conhecimento, simuladores virtuais, quiz interativo, simulados, biblioteca virtual e Explore + para que o aluno possa explorar conteúdos complementares e aprofundar seu conhecimento sobre as temáticas propostas.

8 Temas de aprendizagem

1. PARADIGMAS E LINGUAGEM PYTHON
 - 1.1 LINGUAGENS DE PROGRAMAÇÃO (LP)
 - 1.2 CRITÉRIOS DE AVALIAÇÃO DE LP
 - 1.3 TIPOS DE PARADIGMAS
 - 1.4 IMPLEMENTAÇÃO DAS LP
2. PYTHON BÁSICO
 - 2.1 LINGUAGEM PYTHON
 - 2.2 VARIÁVEIS EM PYTHON
 - 2.3 DADOS E EXPRESSÕES EM PYTHON
 - 2.4 ATRIBUIÇÃO, ENTRADA E SAÍDA DE DADOS EM PYTHON
3. PYTHON ESTRUTURADO
 - 3.1 ESTRUTURAS DE DECISÃO E REPETIÇÃO
 - 3.2 SUBPROGRAMAS EM PYTHON
 - 3.3 BIBLIOTECAS EM PYTHON
 - 3.4 EXCEÇÕES E EVENTOS EM PYTHON
4. PYTHON ORIENTADO A OBJETOS
 - 4.1 ORIENTAÇÃO A OBJETOS (OO)

4.2 PROGRAMAÇÃO OO NA LINGUAGEM PYTHON

4.3 HERANÇA E POLIMORFISMO

4.4 IMPLEMENTAÇÃO DE OO EM PYTHON

5. PYTHON EM OUTROS PARADIGMAS

5.1 LINGUAGEM FUNCIONAL

5.2 COMPUTAÇÃO CONCORRENTE

5.3 PYTHON EM DESENVOLVIMENTO WEB

5.4 PYTHON EM CIÊNCIA DE DADOS

9 Procedimentos de avaliação

Nesta disciplina, o aluno será avaliado pelo seu desempenho nas avaliações (AV ou AVS), sendo a cada uma delas atribuído o grau de 0,0 (zero) a 10,0 (dez). O discente conta ainda com uma atividade sob a forma de simulado, que busca aprofundar seus conhecimentos acerca dos conteúdos apreendidos, realizada online, na qual é atribuído grau de 0,0 (zero) a 2,0 (dois). Esta nota poderá ser somada à nota de AV e/ou AVS, caso o aluno obtenha nestas avaliações nota mínima igual ou maior do que 4,0 (quatro).

Os instrumentos para avaliação da aprendizagem constituem-se em diferentes níveis de complexidade e cognição, efetuando-se a partir de questões que compõem o banco da disciplina. O aluno realiza uma prova (AV), com todo o conteúdo estudado e discutido nos diversos materiais que compõem a disciplina. Será considerado aprovado o aluno que obtiver nota igual ou superior a 6,0 (seis). Caso o aluno não alcance o grau 6,0 na AV, ele poderá fazer uma nova avaliação (AVS), que abrangerá todo o conteúdo e cuja nota mínima necessária deverá ser 6,0 (seis). As avaliações serão realizadas de acordo com o calendário acadêmico institucional.

10 Bibliografia básica

PERKOVIC, Ljubomir. **Introdução à Computação Usando Python - Um Foco no Desenvolvimento de Aplicações**. 1ª Ed. Rio de Janeiro: LTC, 2016.

Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788521630937/>

SEBESTA, Robert W. **Conceitos de Linguagens de Programação**. 11ª Ed. Porto Alegre: Bookkman, 2018.

Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788582604694/>

TUCKER, Allen; NOONAN, Robert. **Linguagens de Programação: Princípios e Paradigmas**. 2ª Ed. Porto Alegre: AMGH, 2014.

Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788563308566/>

11 Bibliografia complementar

AGUILAR, Luis Joyanes. **Programação em C++: Algoritmos, estruturas de dados e objetos**. 2ª Ed.. Porto Alegre: Mc Graw Hill, 2005.

Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788580550269/>

ASCENCIO, Ana F. G.; DE CAMPOS, Edilene A. V. **Fundamentos da Programação de Computadores**. 3º Ed.. Rio de Janeiro: Pearson, 2012.

Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/3272>

MANZANO, José Augusto N. G.; COSTA JR., Roberto A. **Programação de Computadores com Java**. 1ª Ed.. São Paulo: Saraiva, 2014.

Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536519494/>

MENEZES, Alexandre Moreira. **Os Paradigmas de Aprendizagem de Algoritmo Computacional**. 1ª Ed.. São Paulo: Editora Blucher, 2015.

Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788580391039/>

PINHEIRO, Francisco de A. C. **Elementos de Programação em C**. 1ª Ed.. Porto Alegre: Bookman, 2012.

Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788540702035/>