

DISCIPLINA:	LINGUAGENS DE PROGRAMAÇÃO	
Código:	LPR	
Carga Horária Total: 80	CH Teórica: 60	CH Prática: 20
Número de Créditos:	4	
Pré-requisitos:	-	
Semestre:	5º	
Nível:	Superior	
EMENTA		
Aspectos preliminares linguagens, principais conceitos das linguagens, programação funcional, programação lógica.		
OBJETIVO		
Propiciar ao discente os conceitos necessários para a compreensão do funcionamento dos principais paradigmas das linguagens de programação, bem como de suas diferenças e especificidades.		
PROGRAMA		
Unidade I - Aspectos Preliminares. ● Critérios para avaliação de linguagens de programação ● Paradigmas de Programação: imperativas, orientadas a objetos, funcionais e lógicas. Unidade II - Principais Conceitos das Linguagens ● Vinculação e tempo de vida de variáveis ● Escopo de variáveis ● Nomes, Vinculações e Escopos ● Tipos de Dados ● Expressões e Sentenças de Atribuição ● Estruturas de Controle no Nível de Sentença ● Subprogramas ● Tipos de Dados Abstratos Unidade III - Programação Funcional ● Exemplos (Lisp, JavaScript, Perl/Raku) ● Funções puras ● Currificação ● Expressões lambda ● Recursão ● Funções de alta-ordem		

Unidade IV - Programação Lógica

- Fatos e regras
- Unificação
- Backtracking
- Regras recursivas
- Negação por falha

METODOLOGIA DE ENSINO

Aulas teóricas:

- Ministradas em sala, ou outro ambiente que facilite o processo de ensino-aprendizagem, por meio expositivo-dialógico e com discussões com resolução de exercícios, onde a ênfase está em demonstrações conceituais e fundamentos essenciais;
- Como recursos de apoio, tem-se a utilização do quadro branco, projetor de slides e livro(s) de referência(s)

Aulas práticas:

- Ministradas em laboratório de informática, ou outro ambiente que facilite a consolidação dos conceitos fundamentais, por meio do uso e melhoramento de suas habilidades de trabalho ativo, onde a ênfase está na reflexão sobre o que se faz, provocando o encontro de significados no que foi visto na aula teórica.
- Como recursos de apoio, tem-se a utilização de ferramentas para programação de computadores, de plataformas online de ensino aprendizagem de Paradigmas e Linguagens Computacionais e trabalhos dirigidos à utilização de diferentes paradigmas para Linguagens Computacionais

Prática Profissional Supervisionada e projetos interdisciplinares:

- A PPS compreende diferentes situações de vivência profissional, aprendizagem e trabalho, por meio de experiências profissionais supervisionadas pelo professor, onde a ênfase é o estímulo à consolidação de um perfil pró-ativo, com a autoconfiança necessária para uma atuação profissional protagonista
- Deverá ser dada prioridade à realização de projetos interdisciplinares, tais como, por exemplo, o desenvolvimento ações com as disciplinas de Programação Estruturada, POO e/ou compiladores, possibilitando o diálogo entre diferentes disciplinas ou turmas, de maneira a integrar os conhecimentos distintos e com o objetivo de dar sentido a eles.
- Como sugestão de recursos de apoio, tem-se a realização de projetos finais para a disciplina, investigação sobre atividades profissionais, projetos de pesquisa ou outros trabalhos acadêmicos, visitas técnicas, simulações e observações as quais deverão ser desenvolvidas nos diversos ambientes de aprendizagem, como oficinas, incubadoras, empresas pedagógicas ou salas na própria instituição de ensino ou em entidade parceira.

AValiação

O processo avaliativo deve ser contínuo e constante durante todo o processo de ensino-aprendizagem, com o propósito de analisar o progresso do aluno, criando indicadores capazes de apontar meios para ajudá-lo na construção do conhecimento.

Desta forma, para início do processo ensino-aprendizagem, sugere-se avaliações diagnósticas, como forma de se construir um panorama sobre as necessidades dos alunos e, a partir disso, estabelecer estratégias pedagógicas adequadas e trabalhar para desenvolvê-los, inclusive evidenciando os casos que necessitarão de métodos diferenciados em razão de suas especificidades, tais como a necessidade de inclusão. Essas avaliações deverão seguir, preferencialmente, métodos qualitativos, todavia, também seguirão métodos quantitativos quando cabíveis dentro dos contextos individuais e coletivos da turma.

Durante toda a continuidade do processo ensino-aprendizagem, sugere-se a promoção, em alta frequência, de avaliações formativas capazes de proporcionar ao docente um feedback imediato de como estão as interferências pedagógicas em sala de aula, e permitindo ao aluno uma reflexão sobre ele mesmo, exigindo autoconhecimento e controle sobre a sua responsabilidade, frente aos conteúdos já vistos em aula, privilegiando a preocupação com a satisfação pessoal do aluno e juntando informações importantes para mudanças na metodologia e intervenções decisivas na construção de conhecimento dos discentes, inclusive com subsídios para propostas de atividades de recuperação paralela na(s) reunião(ões) de colegiado de curso, coordenadoria de curso e demais setores ligados ao ensino.

Ao final de cada etapa do período letivo, pode-se realizar avaliações somativas, com o objetivo de identificar o rendimento alcançado tendo como referência os objetivos previstos para a disciplina. Há nesses momentos a oportunidade de utilizar recursos quantitativos, tais como exames objetivos ou subjetivos, inclusive com recursos de TIC, todavia, recomenda-se a busca por métodos qualitativos, baseados no planejamento de projetos práticos, práticas interdisciplinares ou atuação em experimentos de laboratório, dentre outros.

BIBLIOGRAFIA BÁSICA

SEBESTA, Robert W. **Conceitos de linguagens de programação**. Porto Alegre: Bookman, 2018. ISBN 9788582604687.

MELO, Ana Cristina Vieira de; SILVA, Flávio Soares Corrêa da. **Princípios de Linguagens de Programação**. São Paulo: Blucher, 2003. ISBN 9788521214922. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/172605>. Acesso em: 19 jul. 2020.

BEN-ARI, M. **Understanding Programming Languages**. Weizmann Institute of Science, 2006. Disponível em: <http://www.weizmann.ac.il/sci-tea/benari/research-activities/understanding-programming-languages>. Acesso em: 27 nov. 2021.

BIBLIOGRAFIA COMPLEMENTAR

AYALA-RINCÓN, Mauricio; DE MOURA, FL Cavalcanti. **Fundamentos da programação lógica e funcional: o princípio de resolução e a teoria de reescrita**. Brasília, DF: Editora UnB, 2014. ISBN 9788523011413.

FERREIRA, Ronaldo. **Linguagem de programação**. Curitiba: Contentus, 2020. ISBN: 9786557450093. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/182483>. Acesso em: 23 nov. 2021.

ARAUJO, Sandro. **Linguagem de programação (ADS)**. Curitiba: Contentus, 2020. ISBN: 9786559350308. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/193067>. Acesso em: 23 nov. 2021.

ROSENFELD, Laurent; DOWNLEY, Allen B. **Think Raku: How to Think Like a Computer Scientist**. Needham, Massachusetts: Green Tea Press, 2020. ISBN: 9781491980552. Disponível em: <https://open.umn.edu/opentextbooks/textbooks/think-raku-how-to-think-like-a-computer-scientist-2nd-edition>. Acesso em: 26 nov. 2021.

KRISHNAMURTHI, Shriam. **Programming languages: applications and interpretation**. 2. ed. Brown University, 2017. Disponível em: <https://open.umn.edu/opentextbooks/textbooks/programming-languages-application-and-interpretation>. Acesso em: 19 jul. 2020.

Coordenador do Curso	Setor Pedagógico