



**UNIVERSIDADE PRESBITERIANA MACKENZIE**  
**Decanato Acadêmico**



<b>Unidade Universitária:</b> Faculdade de Computação e Informática		
<b>Curso:</b> CIÊNCIA DA COMPUTAÇÃO		<b>Núcleo Temático:</b> ALGORITMOS E PROGRAMAÇÃO
<b>Disciplina:</b> PARADIGMAS DE LINGUAGEM DE PROGRAMAÇÃO		<b>Código da Disciplina:</b> ENEX
<b>Professor(es):</b>  DANIELA VIEIRA CUNHA FABIO APARECIDO GAMARRA LUBACHESKI		<b>DRT:</b>  113209-0 114633-0
<b>Carga horária:</b> 4h/a	( 02 ) Sala de Aula ( 02 ) EAD	<b>Semestre Letivo:</b> 1º Semestre/2020
<b>Ementa:</b> Estudo evolutivo das linguagens de programação. Estudo de descrição sintática e semântica de linguagens de programação. Estudo e análise dos principais paradigmas de programação (imperativo, orientado a objetos, funcional, lógico e concorrente).		
<b>Objetivos:</b>		
<b>Fatos e Conceitos</b>	<b>Procedimentos e Habilidades</b>	<b>Atitudes, Normas e Valores</b>
- Reconhecer a pertinência de linguagens a seus respectivos paradigmas. Identificar linguagens multiparadigmas.	- Construir soluções em linguagens de programação, partindo do aprendizado dos conceitos básicos de seu paradigma predominante.	- Ser capaz de decidir, entre paradigmas diferentes, o mais adequado para resolver uma determinada classe de problemas.



**Conteúdo Programático:**

0. Apresentação do Plano de Ensino e dos Pré-Requisitos

0.1. Evolução das linguagens de programação

0.2. Projeto de linguagens de programação: Compiladores, interpretadores e máquinas virtuais

1. Descrição sintática e semântica de uma linguagem de programação

1.1. Análise léxica e sintática

1.2. Nomes, sistemas de tipos, escopo e interpretação semântica

1.3. Gerenciamento de memória

2. Paradigma Imperativo

2.1. Comandos, variáveis, ponteiros

2.2. Abstração Procedural e subprogramas

3. Paradigma Orientado a Objetos

3.1. Objetos, classes, exceções, herança e agregação

3.2. Tipos de polimorfismo. Herança múltipla, classes abstratas e interfaces

3.4. Tipos abstratos de dados

4. Paradigma Funcional

4.1. Cálculo lambda,

4.2. Funções, expressões e listas

5. Paradigma Lógico

5.1. Clausulas de Horn, fatos e regras, backtracking

6. Paradigma de Programação Concorrente

6.1. Conceitos de concorrência (Threads, controle, corridas e deadlocks)

6.2. Estratégias de sincronização (semáforos e monitores)



**Metodologia:**

- Aulas expositivas e participativas utilizando técnicas ativas de aprendizagem
- Lista de exercícios individuais e em grupos
- Utilização do ambiente Mackenzie virtual
- Construção de programas em linguagens representativas de cada paradigma

**Critério de Avaliação:**

**Nota 1 (N1)** composta de:

- Prova parcial 1 (**P1**) escrita, individual e sem consulta (70%) – NOTA A (TIA)
- Atividades de Práticas (**AP1**) (30%) – NOTA B (TIA)

**Nota 2 (N2)** composta de:

- Prova parcial 2 (**P2**) escrita, individual e sem consulta (70%) – NOTA F (TIA)
- Atividades de Práticas (**AP2**) (30%) – NOTA G (TIA)

$$MI = (N1 + N2)/2 + NP$$

- Nota de participação (**NP**): até um ponto (**0 a 0,5**) – atividades definidas pelo professor.

Se  $MI \geq 7.5$ , aluno está APROVADO e a Média Final (MF) = MI.

Caso contrário, poderá fazer uma PROVA SUBSTITUTIVA (SUB), que substituirá a menor nota entre N1 e N2.

Ou, se o aluno ainda não for aprovado, poderá fazer a PROVA FINAL (PF):

$$MF = (MI + PF)/2$$

Se  $MF \geq 6.0$ , aluno está APROVADO. Caso contrário, está REPROVADO.

**Bibliografia Básica:**

SEBESTA, R.W. **Conceitos de Linguagens de programação**. 5ª. ed. Porto Alegre: Bookman, 2011.

SCOTT, M.L. **Programming Language Pragmatics**. 3.ed. Boston: Elsevier, 2009.

TUCKER, A. B.; NOONAN, R. E. **Linguagens de programação: Princípios e Paradigmas**, 2a. ed. Porto Alegre: AMGH, 2014.



**Bibliografia Complementar:**

BRAMER, M. A. **Logic Programming with Prolog**. London: Springer, 2005.

BUDD, T. **An Introduction to Object-Oriented Programming**. 2<sup>nd</sup> ed. Reading: Addison-Wesley, 1998.

MICHAELSON, G. **An introduction to Functional Programming through Lambda Calculus**. New York: Dover, 2011.

SOTTILE, M.; MATTSON, T.G.; RASMUSSEN, C. E. **Introduction to Concurrency in Programming Languages**. Boca Ratón: Chapman & Hall/CRC, 2009.

VAREJÃO, F. M.; GARCIA B. B. **Linguagens de Programação: Conceitos e Técnicas**. Rio de Janeiro: Campus, 2004.