

**Lista 1 de Linguagem de Programação**  
**Acadêmico: Iann Oliveira Silva**

- 1. Conceitue e descreva as diferenças entre (i) linguagem de alto-nível e (ii) linguagem de baixo-nível. Descreva o papel e a importância do compilador no processo de criação de programas de computador.**

R: Uma linguagem de programação de baixo-nível seria aquela que se apresenta mais próxima da forma que o computador representa instruções e dados. A velocidade de execução dos programas e pelo seu tamanho compacto são umas de suas vantagens, exemplo mais famoso o *Assembly*. A linguagem de alto-nível são aquelas que possuem uma estrutura mais fácil de ser compreendida pelo programador, que aproxima o usuário do computador essa é sua maior vantagem para a programação em baixo-nível, exemplos python, perl, C++, C, HTML entre uma grande diversidade cada um com o seu potencial. O compilador é um programa que “lê” um programa escrito em alguma linguagem, e o traduz para a linguagem da máquina, uma importância do compilador é que ele detecta os erros existentes no programa escrito de acordo com a linguagem que esteja sendo executada.

- 2. Faça um programa escrito na linguagem C, C++, Python e Perl que aceite como entrada um número inteiro  $n$  maior ou igual a 1 e retorne como saída o valor da série:**
- 3. Faça um programa escrito na linguagem C, C++, Python e Perl que leia uma quantidade indeterminada de strings na entrada e identifique quais são as duas strings que, respectivamente, são o primeiro e o segundo na ordenação lexicográfica. Exemplo:**
- 4. Faça um programa escrito na linguagem Shell Script para efetuar o download das imagens do tipo .jpg, .jpeg ou .png de um site por meio da utilização de sua URL. Exemplo de execução do programa: ./downscript.sh http://ufrr.br**
- 5. Descreva as seguintes categorias de linguagens de programação e apresente o nome de duas linguagens de programação com seus respectivos exemplos.**

R:

(A) Imperativas : As linguagens imperativas são orientadas a ações, onde a computação é vista como uma sequência de instruções que manipulam valores de variáveis. Um exemplo da muito simples programação imperativa seria: faça isso, depois disso, depois aquilo. Python, C são exemplos de linguagens imperativas.

(B) Funcionais : São as linguagens que se baseiam basicamente em trabalhar com funções, onde ela evita estados e dados mutáveis. Temos Haskell com um exemplo de linguagem funcional.

## **Lista 1 de Linguagem de Programação**

### **Acadêmico: Iann Oliveira Silva**

(C) Lógicas : É a linguagem que trabalho em geral com a lógica matemática, na programação lógica os programas são declarativos: especificam resultados desejados em vez de procedimentos para produzi-los. Um exemplo de linguagem seria o PROLOG.

(D) Marcação/Híbrida : Linguagens de marcação são utilizadas para definir formatos, maneiras de exibição e padrões dentro de um documento qualquer, É representada por um conjunto de códigos empregados a dados e a textos, com a finalidade de adicionar informações específicas sobre esse dado ou texto. A linguagem de marcação é utilizada em diferentes segmentos, em especial na indústria editorial onde a marcação e a formatação de páginas são feitas a partir desse recuso. Um dos melhores exemplos para esse tipo de linguagem seria o HTML.

Programação híbrida seria a que ao mesmo tempo que trabalha com documentos ela pode executar por exemplo funções dentro dos mesmos. Exemplo Java.

- 6. Escreva uma gramática no formato BNF e não ambígua para expressões envolvendo as variáveis A, B, C e os operadores \* (multiplicação) e ^ (exponenciação). A precedência deve seguir as regras usuais da matemática.**

R:

$$\begin{aligned} \langle \text{expr} \rangle &\rightarrow \langle \text{expo} \rangle * \langle \text{expr} \rangle \mid \langle \text{expo} \rangle \\ \langle \text{expo} \rangle &\rightarrow \langle \text{base} \rangle ^ \langle \text{expo} \rangle \mid \langle \text{base} \rangle \\ \langle \text{base} \rangle &\rightarrow A \mid B \mid C \end{aligned}$$

- 7. Descreva o que é um paradigma de programação.**

R:

São as normas orientadoras de um grupo que estabelecem limites e que determinam como um indivíduo deve agir dentro desses limites.

O paradigma de programação pressupõe a forma que programador tem sobre a programação e execução de um programa. Paradigmas diferem nos conceitos e abstrações utilizadas para representar os elementos de um programa (como objetos, funções, variáveis, restrições, etc) e as etapas que compõem um cálculo (atribuição, avaliação, continuações, fluxos de dados, etc.).

Mais claramente falando é a forma de estrutura que você vai usar para programar e executar o seu software, é um conceito que todas as linguagens tem que abordar, porém existem muitos tipos de paradigmas de programação.

- 8. Descreva o que é um analisar léxico. Adicionalmente, apresente uma gramática no formato EBNF para o seguinte programa e exemplifique a sua utilização por meio de uma árvore de derivação.**

## **Lista 1 de Linguagem de Programação**

### **Acadêmico: Iann Oliveira Silva**

R:

O analisador léxico, faz a varredura do programa fonte caractere por caractere e, traduz em uma sequência de símbolos léxicos ou tokens. É nessa fase que são reconhecidas as palavras reservadas, constantes, identificadores e outras palavras que pertencem a linguagem de programação

- 9. Apresente um autômato para reconhecer uma comparação entre números reais. A comparação deve ser feita utilizando os seguintes símbolos: < (menor), > (maior), >= (maior igual), <= (menor igual), = (igual), <> (diferente). Exemplo: 1.2 <= 59.13**

#### **9.1. Defina análise semântica e descreva: Semântica Operacional; Semântica Axiomática; e Semântica Denotacional.**

R:

**Análise semântica:** Análise semântica é a terceira fase da compilação onde se verificam os erros semânticos, no código fonte e coletam-se as informações necessárias para a próxima fase da compilação, que é a geração de código objeto. As validações que não podem ser executadas pelas etapas anteriores devem ser executadas durante a análise semântica a fim de garantir que o programa fonte esteja coerente e o mesmo possa ser convertido para linguagem de máquina.

**Semântica operacional:** Semântica operacional é uma das abordagens de semântica formal, em que o significado de uma construção da linguagem é especificado pela computação que ela induz quando executada em uma máquina hipotética. A semântica operacional preocupa-se mais em como os programas são executados do que meramente com os resultados destas computações.

**Semântica axiomática:** O estilo axiomático presta-se particularmente à prova e raciocínio sobre propriedades dos programas, e à sua verificação, à prova de correção dos programas face às suas especificações, é uma abordagem de semântica formal.

**Semântica denotacional:** Semântica denotacional é um estilo de semântica que visa dar modelos matemáticos para linguagens de programação (por isso ela também é chamada semântica matemática). Significados de programas são definidos como sendo elementos de alguma estrutura matemática apropriada. Um comando de IMP, por exemplo, dado um estado inicial, produz um estado final, ou não termina. Podemos então dizer que comandos de IMP denotam funções parciais entre estados que, por sua vez, são representados como funções de identificadores de variáveis para inteiros.

**Lista 1 de Linguagem de Programação**  
**Acadêmico: Iann Oliveira Silva**

**10. PROGRAMA – A**

$122 * y - 144 > 144$   
 $122 * y > 144 + 144$   
 $122 * y > 288$   
 $y > 288 / 122 \quad ##$

**PROGRAMA – B**

$y = 5 * x - 5$   
 $y + 5 < 45$   
 $y < 40 \quad ##$   
 $5 * x - 5 < 40$   
 $5 * x < 45$   
 $x < 9 \quad ##$

**PROGRAMA – C**

$y > 2$   
 $y + 2 > 2$   
 $y > 0 \quad ##$   
 $y = y - 2$   
 $y - 2 > 2$   
 $y = 4 \quad ##$

**PROGRAMA – D**

Zero iterações

$\{i=N\}$

$wp = (i=i+1, \{i=N\} = \{i+1=N\} \text{ ou } \{i=N-1\})$

entao

$\{i \leq N\}$

**PROGRAMA – E**

$wp=(sn=sn+a, sn=n*a):$

$n*a=sn+a$

$n=sn/a+a/a$

$n=sn/a+1$

$n-1=sn/a$

$a(n-1)=sn$

$\{a=sn/(n-1)\}$