

# Linguagens de Programação e Compiladores

*Avaliação de Linguagens de Programação*

José Osvano da Silva, PMP, PSM I

# Sumário

- › Avaliação de Linguagens de Programação;
- › Critérios para avaliação;
- › Critérios Gerais;
- › Critérios Específicos;
- › Uma Comparação entre C, C++ e JAVA;
- › Considerações Finais.

# Avaliação de Linguagens

- › Linguagens de programação são ferramentas fundamentais para o profissional de computação;
- › Programadores adquirem maior habilidade para resolver problemas e para aprender novas LPs quando têm um bom conhecimento a respeito dos conceitos de linguagens de programação;
- › Mesmo aqueles profissionais que não atuam diretamente como programadores necessitam ter conhecimento a respeito desses conceitos;
- › Esse conhecimento é necessário em todas as etapas do processo de desenvolvimento de software

# Avaliação de Linguagens

- › Ele é importante para analisar a viabilidade do desenvolvimento de uma aplicação;
- › Para estimar o tempo necessário e o custo da implementação
- › Para definir a técnica de projeto de software a ser utilizada
- › E mesmo para aumentar a efetividade na comunicação entre programadores e projetistas de software.

# Avaliação de Linguagens

- › Conhecimento sobre os conceitos de LPs é requisito essencial para realizar uma das tarefas mais fundamentais no processo de desenvolvimento de software:
- › A seleção da linguagem mais apropriada para a implementação de uma aplicação.
- › Nessa direção, vamos apresentar alguns critérios para serem usados na avaliação de LPs com vistas à seleção de linguagens.
- › Esses critérios são utilizados posteriormente para fazer uma comparação entre C, C++ e JAVA.

## Critérios para avaliação

- › A avaliação de linguagens de programação pode ser necessária em várias situações.
- › Ao se desenvolver um sistema computacional, é necessário avaliar as LPs disponíveis para escolher aquela que pode trazer maiores facilidades e benefícios para a elaboração e operação do sistema.
- › Avaliar LPs também é importante para uma instituição no momento de definir ou padronizar as linguagens usadas na implementação de seus projetos.
- › A avaliação de LPs pode ainda ser necessária em comparações técnico-científicas.

## Critérios para avaliação

- › Os critérios escolhidos para avaliação foram separados em dois grupos.
- › No primeiro grupo foram colocados os critérios mais gerais, os quais propiciam uma comparação abrangente das linguagens mas não abordam especificamente os mecanismos e conceitos existentes na LP.
- › O segundo grupo de critérios contém os critérios mais específicos, os quais enfocam fundamentalmente os mecanismos e conceitos oferecidos pelas LPs para possibilitar a implementação de uma ou outra característica em um sistema computacional.

## Critérios Gerais

- › A avaliação dos seguintes critérios gerais deve ser suficiente para definir a LP a ser adotada em um sistema ou por uma instituição em boa parte das situações nas quais isso é necessário.



# CrITÉrios Gerais

Aplicabilidade

Confiabilidade

Facilidade de Aprendizado

Eficiência

# CrITÉrios Gerais

Portabilidade

Suporte ao Método de Projeto

Evolutibilidade

Reusabilidade

# CrITÉrios Gerais

Integração com outros softwares

Custo

# Cr terios Gerais

## Aplicabilidade:

- Avalia se a LP oferece todos os mecanismos necess rios para o desenvolvimento de aplica  es em geral ou de uma aplica  o espec fica.

## Confiabilidade:

- Avalia se a LP promove o projeto e implementa  o de sistemas computacionais confi veis atrav s do uso de conceitos que maximizem a detec   o autom tica de erros e n o estimulem a ocorr ncia de erros.

# CrITÉrios Gerais

## Facilidade de Aprendizado:

- Avalia se a LP oferece uma quantidade regular de conceitos e é simples o bastante para ser facilmente aprendida por um programador.

## Eficiência:

- Avalia o quanto a LP demanda de recursos de memória e processamento durante a execução dos programas

# CrITÉrios Gerais

## Portabilidade:

- Avalia a facilidade para se migrar os códigos fonte dos programas de uma plataforma para outra.

## Suporte ao Método de Projeto:

- Avalia se a LP suporta o método de projeto a ser usado na aplicação específica ou adotado pela instituição.

# CrITÉrios Gerais

## Evolutibilidade:

- Avalia se a LP oferece conceitos adequados para estimular a criação de programas legÍveis e facilmente atualizÁveis.

## Reusabilidade:

- Avalia os meios e facilidades oferecidas pela LP para permitir a reutilização de código.

# Cr terios Gerais

## Integra  o com outros softwares:

- Avalia quais mecanismos a LP oferece (ou necessita) para constru  o de programas que incorporem (ou sejam incorporados) por programas implementados em outras LPs

## Custo:

- Avalia o custo financeiro necess rio para o uso da linguagem, para a aquisi  o das ferramentas de desenvolvimento (tais como, editores de programas, compiladores, depuradores) na plataforma na qual ser  desenvolvida a aplica  o e para uso da aplica  o
- (tal como, por exemplo, se existe custo para executar a aplica  o utilizando uma m quina virtual).



## Critérios Específicos

- › Esse grupo de critérios é mais apropriado para realizar comparações técnico-científicas entre LPs;
- › Alguns deles, em algumas situações, também possam ser importantes na definição da LP adotada em um sistema ou por uma instituição.
- › Para não tornar o processo de comparação muito extenso, optou-se por selecionar apenas critérios considerados mais relevantes.
- › Os critérios específicos são listados a seguir.

# Critérios Específicos

Escopo

Expressões e Comandos

Tipos Primitivos e Compostos

Gerenciamento de Memória

# Critérios Específicos

Persistência de Dados

Passagem de Parâmetros

Encapsulamento e Proteção

Sistema de Tipos

# Critérios Específicos

Verificação de Tipos

Polimorfismo

Exceções

Concorrência

# Critérios Específicos

## Escopo:

- Avalia se a LP requer a definição (ou declaração) explícita das entidades de programação, associando-lhes um escopo de visibilidade determinado pela organização textual do programa

## Expressões e Comandos:

- Avalia se a LP oferece uma boa variedade de expressões e/ou comandos para a construção de programas estruturados.

# Critérios Específicos

## Tipos Primitivos e Compostos:

- Avalia se a LP oferece uma ampla variedade de tipos primitivos e compostos, permitindo representar qualquer categoria importante de dado. Complementarmente, esse critério deve levar em conta a forma de tratamento do tipo String na LP.

## Gerenciamento de Memória:

- Avalia se a LP oferece um mecanismo próprio para gerenciamento de memória ou se o deixa sob a responsabilidade do programador.

## Critérios Específicos

### Persistência de Dados:

- Avalia o suporte oferecido pela LP para a realização de operações relacionadas com a persistência de dados.

### Passagem de Parâmetros:

- Avalia os modos e mecanismos disponíveis na LP para realização da passagem de parâmetros.

# Critérios Específicos

## Encapsulamento e Proteção:

- Avalia se a LP oferece mecanismos para encapsulamento e proteção dos dados.

## Sistema de Tipos:

- Avalia se a LP impõe (ou não) uma disciplina rigorosa na realização de operações sobre os dados, coibindo a execução de operações sobre dados de tipos para os quais não foi planejada e impedindo a manipulação de dados de um determinado tipo como se fossem de outro.



# Critérios Específicos

## Verificação de Tipos:

- Avalia se a verificação de tipos das operações é feita estática ou dinamicamente.

## Polimorfismo:

- Avalia os tipos de polimorfismos oferecidos pela LP.
- Em caso de existência de polimorfismo de inclusão, é importante avaliar se a LP oferece herança simples ou múltipla.

# Critérios Específicos

## Exceções:

- Avalia se a LP oferece mecanismos específicos para tratamento de exceções ou se o controle de erros é deixado sob responsabilidade exclusiva dos programadores.

## Concorrência:

- Avalia os recursos oferecidos pela LP para a construção de programas concorrentes.

# Dúvidas



**José Osvano da Silva**  
[joseosvano@unipac.br](mailto:joseosvano@unipac.br)