

Linguagens de Programação

Jonas Alves Bueno¹ Antônio Egydio²

^{1,2} FATEC

jonas.bueno@fatec.sp.gov.br antonio.graca@fatec.sp.gov.br

1. Introdução

Ao estudar sobre o funcionamento dos computadores aprendemos que são máquinas que, em última instância, só compreendem a presença ou ausência de sinais elétricos, representados por “zeros” e “uns”, ou bits. Porém atualmente podemos executar tarefas extremamente sofisticadas com apenas alguns cliques ou algumas linhas de código. Para entender como isso é possível, vamos investigar sobre as Linguagens de Programação.

2. Definição e História

Toda linguagem consiste em um “acordo”, ou uma padronização que conecta determinados símbolos à significados correspondentes. Nas linguagens de programação não é diferente: elas nada mais são que um conjunto de palavras-chave que *abstraem* instruções de máquina muito mais complexas se analisadas no nível da linguagem de máquina (os zeros e uns). Também assim como as linguagens naturais, as LP (Linguagens de programação) seguem regras semânticas e sintáticas.

Nos primórdios dos computadores eletrônicos não existiam sistemas operacionais ou LP's. Toda a programação dessas máquinas era feito manualmente através de cartões perfurados, onde a ausência ou presença de buracos era lido como “zero” ou “um”. O resultado disso eram programas como o representado na *Figura 1*: Quilos de cartões perfurados para produzir programas simples. Se um deles fosse inserido fora de ordem ou uma perfuração fosse feita no lugar errado, todo o programa seria comprometido e achar o erro poderia demorar dias.

As LP's possuem ser classificadas em relação à *como* o conjunto de instruções ou algoritmo é transformado em linguagem de máquina para ser compreendido e executado pela máquina. Neste aspecto, existem linguagens *compiladas* e *interpretadas*. No primeiro caso, após finalizado o código, o programa passa por um **compilador** que “traduz” as instruções para linguagem da máquina, produzindo o código fonte que pode ser agora executado indefinidamente. Exemplos de linguagens compiladas são C e Java. Nas linguagens interpretadas (Python, JavaScript e Ruby, por exemplo), um **interpretador** traduz e executa o programa linha a linha, sem a necessidade de gerar um binário.

As LP's também são classificadas como de **alto** ou **baixo nível**. Esta classificação indica se a LP é mais próxima à linguagem de máquina (baixo nível) ou à linguagem natural (alto nível – em geral tendo o inglês como a linguagem natural de referência).

Além das vantagens de aumento da produtividade e diminuição da curva de aprendizado das linguagens de

alto nível em relação à linguagens como *assembly* (linguagem de baixo nível usada para programação de microprocessadores), um outro ponto importante é o da **portabilidade**. Ou seja, os programadores se tornam menos dependentes de máquinas específicas e podem escrever códigos uma vez e rodá-los em qualquer lugar.



Figura 1 – Programadora junto à um pilha de cartões perfurados que correspondiam à um programa de 5 Megabytes.

3. Conclusões

Entender a fundo o funcionamento das linguagens de programação, bem como a lógica de programação, consistem em campos de estudo fundamentais e incontornáveis para os profissionais de tecnologia da informação, pois são a principal interface de comunicação homem/máquina.

4. Referências

- [1] EGYDIO, Antônio. **Linguagens de Programação**
- [2] Marçula, M., Filho, P: **Informática: Conceitos e aplicações**. Érica, 2008, São Paulo.

¹ Discente do curso de Administração e Desenvolvimento de Sistemas na instituição Fatec, *campus* São José dos Campos.

² Docente da disciplina “Arquitetura e Organização de Computadores” na instituição Fatec - *campus* São José dos Campos.