Universidade Presbiteriana Mackenzie

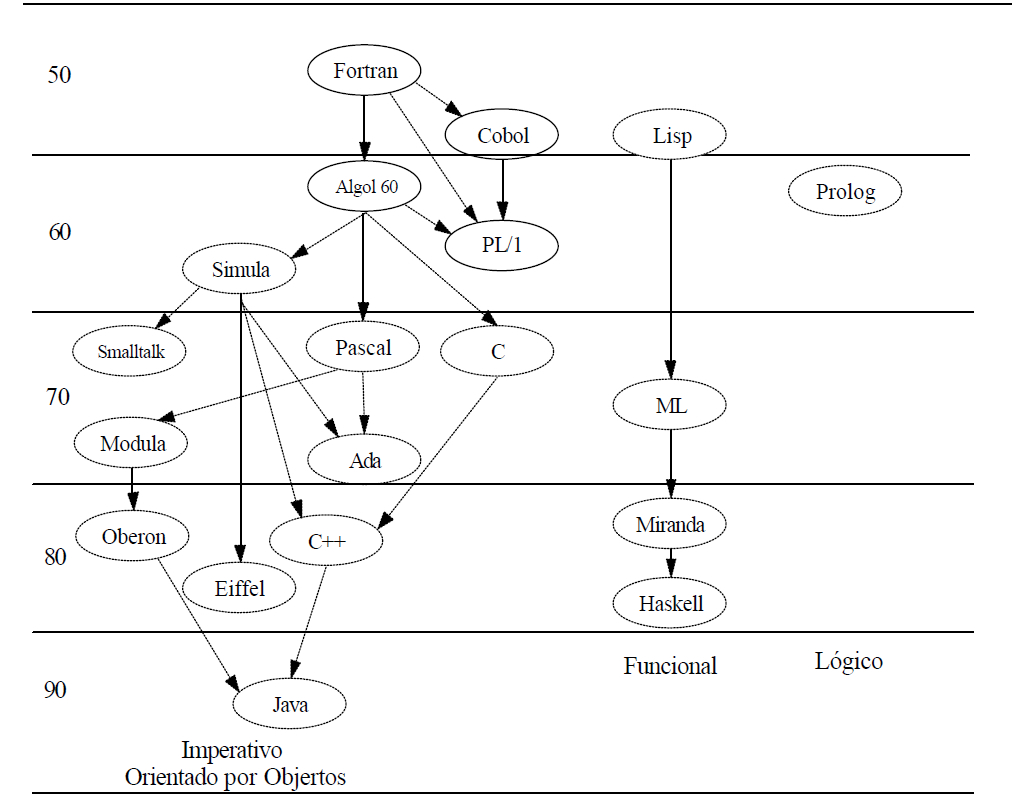
Matéria: Paradigmas de Linguagem de Programação

**CONCEITOS INTRODUTÓRIOS**

Ricardo Arthur de Souza Simião - 31676987

Clauber Pereira Stipkovic Halic - 31243045

**1. EVOLUÇÃO DAS PRINCIPAIS LINGUAGENS DE PROGRAMAÇÃO**



As primeiras Linguagens de programação foram as linguagens de maquina e as linguagens assembly dos primeiros computadores, começando na década de 1940. Centenas de linguagens de programação e dialetos foram desenvolvidos desde então. A maioria teve seu tempo de vida e utilidade limitados e algumas tiveram amplo sucesso em um ou mais domínio de aplicações, muitos desempenharam importante influência sobre o projeto de futuras linguagens.

Começando com a **Plankalkül** que criada pelo cientista alemão Konrad Zuse, entre os anos 1942-1946 no desenvolvimento dos primeiros computadores. A linguagem é considerada de alto nível, mas só foi publicada amplamente em 1972, e o compilador em 1998. Uma implementação independente foi lançada em 2000 pela Universidade Livre de Berlim. De acordo com Heinz Rutishauser (co-fundador do **ALGOL**), foi a primeira tentativa de desenvolver uma linguagem algorítmica; Sua notação foi geral , mas a proposta nuca alcançou a consideração que merecesse.

A década de 1950 marcou o inicio da era das Linguagens de ordem mais alta ou linguagem de alto nível, pois são independente das arquitetura de máquinas específicas. Muito utilizada na época, a linguagem **FORTRAN** foi uma das impulsionadoras dessa geração e originou o **COBOL** que é mais voltado para aplicações comerciais, essa muito utilizada até hoje.

Logo após o **FORTRAN** veio o **ALGOL** , criado na década de 60 foi a pioneira em demonstrar os conceitos de uma linguagem moderna trazendo os conceitos de estrutura de bloco e procedimentos, foi a primeira linguagem de programação estruturada e criou desafios técnicos para a construção de compiladores ( Nenhuma linguagem teve influência tão importante nos projetos de novas linguagens). Outra importante foi a **SIMULA** que deu origem à linguagem **Smaltalk** , lançada na década de 70 , que certamente é a única linguagem puramente orientada a objetos.

O Pascal é a linguagem mais popular de todas descendentes de Algol. Simples sistemáticas e com implementações eficientes, pensando em propagar rapidamente o uso de linguagem , foi criado , em Zurique , um “Kit de Conversão” que incluía um compilador que gerava código intermediário , e um simulador para ele e foi utilizado para diversas coisas entre elas a criação de um sistema operacional que foi instrumental para a popularização do PASCAL, foi também o carro-chefe da Borland em 1980 com o TURBO PASCAL.

Durante esse tempo surgia o **LISP** , que entre sua concepção , implementação e seus primeiros dialetos foram 9 anos de esforços empreendidos. Apesar das primeiras implementações do LISP terem sido realizados nos IBM 704 e 7090, trabalhos posteriores concentraram-se nos DEC PDP-6 e PDP-10, este último sendo o baluarte do Lisp e das pesquisas em IA (inteligência artificial) em lugares como o MIT (Massachussets Institute of Tecnology) e as Universidades de Stanford e Carnegie-Mellon até metade dos anos 1970. O computador PDP-10 e seu antecessor, o PDP-6 eram por definição, especialmente adequados para o LISP, por possuírem palavras de 36 bits e endereços de 18 bits. Esta arquitetura permitia um registro de um cons cell (par pontuado) em uma única palavra de memória, em instruções simples extraíam o seu car e cdr. Esses computadores possuíam também poderosas instruções de pilha, que proporcionavam rápida chamada a funções; porém suas limitações em 1973 eram evidentes: suportavam um pequeno número de pesquisadores utilizando o LISP e seu endereçamento em 18 bits limitava o espaço dos programas. Uma resposta para o problema de endereçamento foi o desenvolvimento do "LISP Machine",um computador dedicado especialmente à tarefa de trabalhar com a linguagem. Outra solução foi a utilização de computadores de uso geral com maior capacidade de endereçamento, como o DEC VAX e o S1 Mark IIA. Até 1980, LISP se tornou a principal linguagem da comunidade de inteligência artificial, tendo sido pioneiro em aplicações como administração automática de armazenamento , linguagens interpretadas e programação funcional. Atualmente os dialetos mais conhecidos são : Common LISP , Scheme e Cloujure.

# **2. LINGUAGENS MÃES**

Fortran, SMALLTALK , C , LISP

# **3. POR QUE ALGUMAS LINGUAGENS DEIXAM DE EVOLUIR?**

Pela rápida evolução da demanda de poder computacional e porque elas já não atendem mais os domínios de problemas de uma certa comunidade com suas aplicações.

# 4. POR QUE ALGUMAS LINGUAGENS SEMPRE EVOLUEM?

A diferença está em um conjunto de características chaves chamado de objetivos de projeto, o que é levado em conta é sua simplicidade, clareza nas ligações , confiabilidade , suporte , Abstração , Ortogonalidade e Implementação eficiente.

**SIMPLICIDADE:**

As Linguagens de sucesso devem ser compreensíveis e fáceis de ler pelo programador mediano

**CLAREZA NAS LIGAÇÕES:**

Diz respeito o quão um elemento está conectado a uma propriedade quando essa propriedade é definida para ele. Uma boa linguagem deve ser muito clara sobre quando ocorre a ligação principal para cada elemento com sua propriedade (Tempo de definição. Tempo de implementação de linguagem , tempo de escrita, tempo de compilação , tempo de carga do programa, tempo de execução do programa).

**CONFIABILIDADE:**

O programa se comporta da mesma forma toda vez que ele é executado em diferentes plataformas? O Seu comportamento pode ser especificado independentemente de modo a permitir a sua verificação formal (ou informal).

Particularmente pertinente a essas questões é a necessidade de se projetar mecanismos apropriados de manipulação de excessões na linguagem. Além disso , linguagens restrinjam o uso de aliases e vazamento de memória , que suportem tipagem forte, tenha sintaxe e semântica bem definidas e que suportem a verificação e validação dos programas têm uma vantagem nessa categoria.

**SUPORTE:**

Uma boa linguagem de programação deve ser facilmente acessível por alguém que queira aprendê-la e instalá-la no seu próprio computador. De forma ideal , seus compiladores devem ser de domínio publico , em vez de serem propriedade de uma corporação e caros de obter. A linguagem devem ser implementadas em múltiplas plataformas. Cursos, livros , tutoriais e um grande numero de pessoas familiarizadas com a linguagem ajudam a preservar e a estender a vitalidade da linguagem.

**ABSTRAÇÃO:**

A Abstração é um aspecto fundamental do processo de projeto de programas. Os programadores gastam muito tempo construindo abstrações, tanto de dados quanto procedurais, para explorar a reutilização de código e evitar reinventá-lo Uma ferramenta de programação suporta tão bem a abstração de dados e procedural que é a ferramenta de projeto preferida na maioria das aplicações.

Bibliotecas que acompanham linguagens modernas de programação comprovam a experiência acumulada de programadores na construção de abstrações.

**ORTOGONALIDADE:**

Uma linguagem é dita ortogonal se os seus comandos e recursos são construídos sobre um conjunto pequeno e mutuamente independente de operações primitivas. Quanto mais ortogonal uma linguagem , menos regras excepcionais são necessárias para escrever programas corretos. Assim programas em uma linguagem ortogonal muitas vezes tendem a ser mais simples e mais claros do aqueles em uma linguagem não ortogonais.

**IMPLEMENTAÇÃO EFICIENTE:**

Os recursos a as construções de uma linguagem devem permitir uma implementação prática e eficiente em plataformas contemporâneas.

# 5. AS LINGUAGENS MAIS UTILIZADAS HOJE (TIOBE SOFTWARE). JUSTIFIQUE ESSA CLASSIFICAÇÃO

É uma lista ordenada de linguagens de programação, classificada pela freqüência de pesquisa na web usando o nome da linguagem como a palavra-chave.

O índice cobre buscas no Google, Google Blogs, MSN, Yahoo!, Wikipedia e no YouTube. O índice é atualizado mensalmente. A informação mais recente é gratuita, mas as análises e estatísticas de dados ao longo dos anos é paga. Os autores do índice acham que essas informações podem ser valiosas ao aceitar variadas decisões estratégicas. O TIOBE foca em linguagens de programação completas (Turing completa), logo, não fornece informações sobre a popularidade de, por exemplo, SQL ou HTML.

# 6. BIBLIOGRAFIA

Tucker, Allen B. Nooman, R. Linguagens de Programação - Princípios e Paradigmas - 2ª Ed, Amgh Editora

TIOBE Index for August 2016 - http://www.tiobe.com/tiobe-index/ (visitado em 22/08/2016)

Imagem - https://widegroup.files.wordpress.com/2010/12/lps.jpg (visitado em 20/08/2016)