

Linguagens de Programação: História das Linguagens

Prof. Arnaldo Candido Junior

UNESP – IBILCE

São José do Rio Preto, SP

Popularidade das Linguagens

- Neste tópicos, discutiremos a evolução das linguagens de programação
- Mas antes, vamos dar uma olhada nas linguagens mais populares
- Existem rankings que medem a popularidade das linguagens segundo diferentes métricas
- Veremos três rankings

Popularidade das Linguagens (2)

- 1. Tiobe

<https://tiobe.com/tiobe-index/>

- 2. RedMonk

<https://redmonk.com/sogrady/2022/10/20/language-rankings-6-22/>

<https://redmonk.com/rstephens/2021/08/05/top-20-june-2021/>

- 3. PYPL

<https://pypl.github.io/PYPL.html>

- 4. StackOverflow

<https://survey.stackoverflow.co/2022/#most-loved-dreaded-and-wanted-language-love-dread>

Exercício: três pontos chaves

- Descrever **três pontos** chaves / curiosidades / coisas legais de uma linguagem do ranking Tiobe
- Para alunos presentes na aula, o exercício poderá ser feito em dupla
- Para alunos entregando a atividade posteriormente, o exercício é individual
- A linguagem recebida será feita por sorteio

Exercício: estruturas

- Para a linguagem escolhida, mostrar quatro programas
 - 1. “Hello World”
 - 2. Subprograma para calcular a média entre dois valores (escolher inteiros ou reais)
 - 3. If para checar se um número é par ou ímpar
 - 4. Laço (ou recursão) para contar de 1 até 10

Exercício: estruturas (2)

- Enviar print da tela mostrando os 4 programas rodando
- Pode ser: Google Docs; documento Office; documento LibreOffice; ou um zip com imagens
- Usar compiladores online ou local para compilar a linguagem escolhida
- Se não encontrar um compilador disponível, fazer a atividade com a linguagem seguinte do ranking Tiobe

Exercício: estruturas (3)

- Compiladores online
 - <https://replit.com/>
 - <https://rextester.com/>
 - <https://www.tutorialspoint.com/codingground.htm>
 - <https://www.onlinegdb.com/>
 - <https://www.programiz.com/>

Computação antes das linguagens

- Ábaco (séc. III d.C.): usa discos para efetuar cálculos
- Pascalina (1642): operações aritméticas simples por discos interligados
- Calculadora de Leibniz (1673): trazendo multiplicação e divisão
- Tear de Jacquard (1802): cartões perfurados lidos em uma máquina de tear

Computação antes das linguagens (2)

- Máquina Diferencial (1822) de Charles Babbage: calculava funções trigonométricas e logarítmicas
- Máquina Analítica (1833/1834) de Charles Babbage: executava as quatro operações aritméticas e trazia o conceito de memória

Ada Lovelace

- Considerada pelos historiadores como a primeira programadora
- Interessou-se pelo trabalho de Babbage, a máquina máquina analítica em 1834
- Ao traduzir um artigo de francês para inglês adicionou várias notas que mostraram que ela entendeu o funcionamento da máquina

Ada Lovelace (2)

- Conceito de procedimento: sequência de cartões (programa) independente dos valores operados
- Noção de símbolos e variáveis de memória
- Noção da atribuição de valores a variáveis
- Escreveu uma sequência - como calcular os números de Bernoulli - é considerado pelos historiadores da computação como o primeiro programa de computador (1843)

Ada Lovelace (3)

- Foi homenageada por uma linguagem de mesmo nome na década de 80
- <https://www.celebratingada.com/>

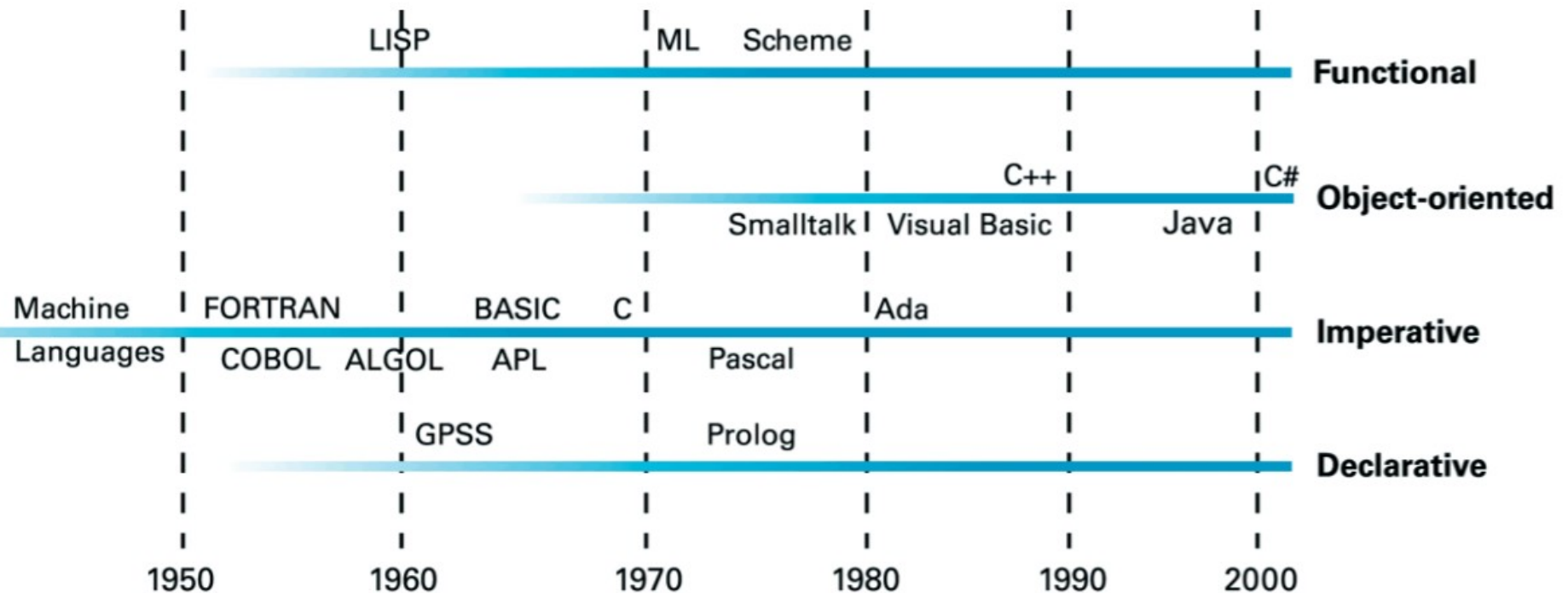
Ada Lovelace (4)

Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 et seq.)

Number of Operations.	Nature of Operation.	Variables acted upon.	Variables receiving results.	Indication of change in the value on any Variable.	Statement of Results.	Data.												Working Variables.												Result Variables.					
						$1V_1$	$1V_2$	$1V_3$	$2V_4$	$2V_5$	$2V_6$	$2V_7$	$2V_8$	$2V_9$	$2V_{10}$	$2V_{11}$	$2V_{12}$	$2V_{13}$	$2V_{14}$	$2V_{15}$	$2V_{16}$	$2V_{17}$	$2V_{18}$	$2V_{19}$	$2V_{20}$	$2V_{21}$	$2V_{22}$	$2V_{23}$	$2V_{24}$	$2V_{25}$	$2V_{26}$	$2V_{27}$	$2V_{28}$		
						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
						1	2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
						1	2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
1	\times	$1V_2 \times 1V_3$	$2V_6, 1V_9, 1V_8$	$1V_6 = 1V_2$ $1V_9 = 1V_3$ $1V_8 = 1V_2 \times 1V_3$	$-2n$...	2	n	2n	2n	2n																								
2	$-$	$1V_4 - 1V_2$	$2V_5$	$2V_5 = 1V_4 - 1V_2$	$-2n-1$	1	2n-1																										
3	$+$	$1V_5 + 1V_1$	$2V_6$	$2V_6 = 1V_5 + 1V_1$	$-2n+1$	1	2n+1																										
4	$+$	$2V_8 + 2V_4$	$2V_{11}$	$2V_{11} = 2V_8 + 2V_4$	$-\frac{2n-1}{2}$	0	0																										
5	$+$	$2V_{11} + 1V_3$	$2V_{12}$	$2V_{12} = 2V_{11} + 1V_3$	$-\frac{1}{2} \cdot \frac{2n-1}{2n+1}$...	2																										
6	$-$	$2V_{12} - 1V_{11}$	$2V_{13}$	$2V_{13} = 2V_{12} - 1V_{11}$	$-\frac{1}{2} \cdot \frac{2n-1}{2n+1} = A_2$																										
7	$-$	$1V_3 - 1V_1$	$2V_{10}$	$2V_{10} = 1V_3 - 1V_1$	$= n-1 (-3)$	1	...	n	...																										
8	$+$	$1V_2 + 2V_7$	$2V_7$	$2V_7 = 1V_2$ $2V_7 = 1V_2$	$-2+0-2$...	2																										
9	$-$	$2V_8 - 2V_7$	$2V_{14}$	$2V_{14} = 2V_8 - 2V_7$	$-\frac{2n}{2} = A_1$																										
10	\times	$2V_{12} \times 2V_{10}$	$2V_{15}$	$2V_{15} = 2V_{12} \times 2V_{10}$	$= B_2 \cdot \frac{2n}{2} = B_2 A_1$																										
11	$+$	$2V_{15} + 2V_{14}$	$2V_{16}$	$2V_{16} = 2V_{15} + 2V_{14}$	$-\frac{1}{2} \cdot \frac{2n-1}{2n+1} + B_2 \cdot \frac{2n}{2}$																										
12	$-$	$1V_{13} - 1V_1$	$2V_{19}$	$2V_{19} = 1V_{13} - 1V_1$	$= n-2 (-2)$	1																										
13	$-$	$1V_6 - 1V_1$	$2V_6$	$2V_6 = 1V_6 - 1V_1$	$-2n-1$	1																										
14	$+$	$1V_1 + 2V_7$	$2V_7$	$2V_7 = 1V_1$ $2V_7 = 2V_7$	$-2+1=3$	1																										
15	$-$	$2V_8 - 2V_7$	$2V_8$	$2V_8 = 2V_8 - 2V_7$	$-\frac{2n-1}{2}$																										
16	\times	$2V_8 \times 2V_{10}$	$2V_{13}$	$2V_{13} = 2V_8 \times 2V_{10}$	$-\frac{2n}{2} \cdot \frac{2n-1}{2}$																										
17	$-$	$2V_8 - 1V_1$	$2V_8$	$2V_8 = 2V_8 - 1V_1$	$-2n-2$	1																										
18	$+$	$1V_1 + 2V_7$	$2V_7$	$2V_7 = 1V_1$ $2V_7 = 2V_7$	$-3+1=4$	1																										
19	$-$	$2V_8 - 2V_7$	$2V_8$	$2V_8 = 2V_8 - 2V_7$	$-\frac{2n-2}{4}$																										
20	\times	$2V_8 \times 1V_{11}$	$2V_{13}$	$2V_{13} = 2V_8 \times 1V_{11}$	$-\frac{2n}{2} \cdot \frac{2n-1}{2} = A_2$																										
21	\times	$2V_{12} \times 2V_{11}$	$2V_{13}$	$2V_{13} = 2V_{12} \times 2V_{11}$	$= B_2 \cdot \frac{2n-1}{2} = B_2 A_2$																										
22	$+$	$2V_{12} + 2V_{13}$	$2V_{13}$	$2V_{13} = 2V_{12} + 2V_{13}$	$= A_2 + B_2 A_1 + B_2 A_2$																										
23	$-$	$2V_{13} - 1V_1$	$2V_{13}$	$2V_{13} = 2V_{13} - 1V_1$	$= -3 (-1)$	1																										

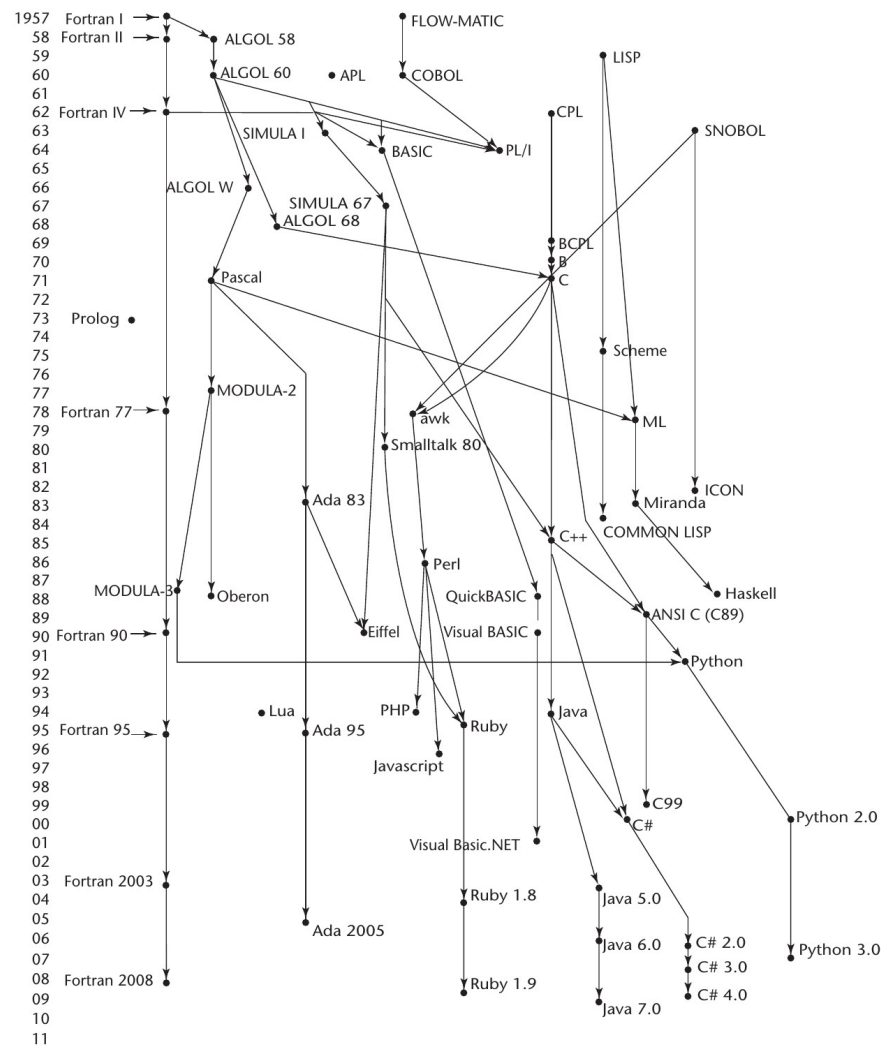
Here follows a repetition of Operations thirteen to twenty-three.

História das Linguagens



Fonte: https://en.wikipedia.org/wiki/Timeline_of_programming_languages

História das Linguagens (2)



História das Linguagens (3)

- Há divergência nas datas de criação de algumas linguagens
 - Alguns autores citam o projeto e a concepção
 - Outros a publicação do primeiro compilador
 - Seguiremos as definições do livro texto

Plankalkul

- Proposta em 1946 por Konrad Zuse
 - Construtor das máquinas Z1, Z2, Z3
(calculadores mecânicos de ponto flutuante)
- Linguagem que permitiria um nível de abstração mais alto
- Não chegou a ser implementada até 1975

Plankalkul (2)

- Considerada por Zuze como um exercício mental
- Primeira concepção de um compilador: leria comandos nessa linguagem e automaticamente perfuraria cartões com os comandos em linguagem de máquina

Plankalkul (3)

subrotina

Plankalkül

atribuição

```
P1 max3 (V0[:8.0],V1[:8.0],V2[:8.0]) → R0[:8.0]
max(V0[:8.0],V1[:8.0]) → Z1[:8.0]
max(Z1[:8.0],V2[:8.0]) → R0[:8.0]
END
P2 max (V0[:8.0],V1[:8.0]) → R0[:8.0]
V0[:8.0] → Z1[:8.0]
(Z1[:8.0] < V1[:8.0]) → V1[:8.0] → Z1[:8.0]
Z1[:8.0] → R0[:8.0]
END
```

condicional

Década de 1940

- Computadores ficam mais avanços, baseados em válvulas
- 1943: Mark I, o primeiro computador eletromecânico
- 1943: criado para quebrar códigos alemães ultra-secretos.
- 1946: ENIAC - Computador e Integrador Numérico Eletrônico
- A programação é feita em código de máquina

Década de 1950

- Nesta década, válvulas começam a ser substituídas por transistores
- Surgem as primeiras linguagens de programação de alto nível.
- Entre elas, Fortran, Cobol, Lisp
 - Precursoras dos paradigmas Imperativo, Funcional e Lógico
 - Nichos diferentes

Década de 1950 ⁽²⁾

- Surgem as primeiras linguagens de montagem, a época conhecidas como pseudocodes
- Surgem dois paradigmas, em forma ainda rudimentar
 - Imperativo: Fortran, Cobol, Algol 58
 - Funcional: Lisp
- Aplicações iniciais: classificação de padrões, sistemas especialistas , processamento de língua natural, entre outras

Pseudocodes

- Fim de 1940: as primeiras linguagens de montagem eram conhecidas como pseudocodes
- A palavra evoluiu e não tem mais o mesmo significado que na época
- Exemplos: short code; speedcoding; Univac “compiling”
- Detalhes no livro texto

Fortran

- Foco focu em aplicações científicas: realização de cálculos matemáticos complexos com baixo custo para o computador
- Aplicado na engenharia e em aplicativos científicos para a resolução de cálculos matemáticos amplos.
- Introduziu conceitos como variáveis, vetores, estruturas de seleção e estruturas de repetição primitivas.

Cobol

- Foco aplicações comerciais. Permitia produção de relatórios elaborados
- Aplicado no gerenciamento empresarial para realizar análises de dados ou controle de finanças
- Expandiu os conceitos com mais variáveis decimais, estruturas aninhadas e menos foco em eficiência
- Predecessora das planilhas eletrônicas e dos Sistemas de Gerenciamento de Bancos de Dados (SGBD)

LISP

- Foco aplicações de Inteligência Artificial
- Computação simbólica (ao invés da numérica)
- Introduziu manipulação de informações utilizando listas e não vetores
- Manipulação de listas por meio de recursão

Década de 1960

- Período foi marcado pela evolução e refinamento das linguagens de programação
- Surge a linguagem Algol 60, intensamente debatida, mas acabou não se popularizando
 - Influenciou outras linguagens como C e Pascal
- Simula 67: extensão do Algol 60 com foco em abstração de dados
 - Conceito usado posteriormente para a criação das linguagens orientadas à objetos

Década de 1960 ⁽²⁾

- PL/I (Programming Language I): proposta pela IBM
 - Primeira tentativa em larga escala para a criação de uma linguagem multi-nicho
 - Acabou não atingindo a popularidade esperada
- Surge a primeira versão da linguagem Basic
- No final da década surge a linguagem C

Década de 1970

- Paradigma funcional continua evoluindo com Scheme e Common Lisp sendo derivadas do Lisp
- Paradigma lógico surge com a criação da linguagem Prolog
- Paradigma orientado a objetos começa a tomar forma. Na virada da década, surge a linguagem Smalltalk
- Paradigma imperativo se consolida como o mais popular: Algol 68, Basic, Pascal, C

Década de 1970 ₍₂₎

- Pascal: foco em simplicidade
- Linguagens orientadas a usuário: apoiam o programador a desenvolver algoritmos
 - Tipos de dados bem definidos (int, float, char).
 - Estruturas de Seleção (if) e Repetição (for, while)
 - Tipos Abstratos de Dados

Linguagem C ₍₂₎

- Projetada por Dennis Richie, em 1972
- Linguagem de sistemas portátil
- Grande quantidade de rotinas pré-compiladas em bibliotecas disponível para programadores
- Linguagem de médio nível (combina alto e baixo nível)

Linguagem C ₍₂₎

- Alto nível em C: elementos de uma linguagem imperativa tradicional
 - Tipagem de Dados, criação de variáveis, estruturas de dados, estruturas de controle, entrada e saída de dados.
- Baixo Nível em C: instruções em Assembly para otimizar desempenho do código.

Linguagem C ₍₃₎

- Principais Aplicações:
 - Ensino de Programação
 - Sistemas multi-plataforma
 - Jogos de computador
 - Sistema Operacional Unix
 - Base dos Sistemas Operacionais Modernos: Windows, Linux, IOS, Android.

Linguagem C (4)

- A linguagem C é predecessora de várias linguagens de programação:
 - C#
 - Java
 - JavaScript
 - Entre varias outras

BASIC

- Beginner's All-purpose Symbolic Instruction Code
- Foco: fácil para estudantes que não são de computação
- Prioriza o tempo do usuário mais que o tempo da máquina
 - Estratégia acertada considerando-se que a computação se tornaria mais barata
- Muito popular no final da década de 70

Prolog

- Usa notação lógica como forma de se comunicar com a máquina
 - Oficialmente, é a mesma notação da área de Cálculo de Predicados
- Abordagem: não focar em **como** resolver o problema. Foca em definir **o que** é o problema
 - Cabe a máquina como resolvê-lo

Década de 1980

- Foco:
 - Abstração (Funções, Subprogramas).
 - Correção de programas: verificação de tipos, exceções.
 - Programação concorrente e distribuída em tempo real.
 - Tipos Abstratos de Dados (TADs).

Década de 1980 ⁽²⁾

- Consolidação do Paradigma Orientado a Objetos: classe e objeto; polimorfismo; encapsulamento; herança
- Duas linguagens dinâmicas pioneiras são propostas: APL e Snobol

Década de 1980 ⁽³⁾

- Evolução de linguagens dos anos 60 e 70:
 - C++ (Originado da linguagem C)
 - Object Pascal (Originado da linguagem Pascal)
 - Common Lisp (Originado da linguagem Lisp)

SmallTalk

- Primeira linguagem a oferecer suporte completo à programação OO
- Tudo na linguagem é tratado como objetos, mesmo constantes inteiras
- Traz o conceito de classes originalmente proposto pela linguagem Simula 67

C++

- Desenvolvida nos laboratórios Bell por Bjarne Stroustrup, em 1985
 - Utilizou a estrutura da linguagem C e adicionou os principais elementos de Orientação a Objeto
 - Principais Modificações: Classe, Encapsulamento, Herança, Tratamento de Exceções.
 - Código C pode ser mesclado com código C++

C++ (2)

- Híbrida, permitindo programação Procedural e Orientada a Objetos
- Principais Aplicações:
- Produtividade: office, editores de imagens, de vídeos
- Entretenimento: Jogos
- Navegadores: Internet Explorer (Microsoft), Google Chrome (Google), Mozilla Firefox

ADA

- Grande esforço para projeto de uma linguagem de programação poderosa
 - Teve por traz o Departamento de Defesa dos EUA
- Gerou muita atenção quando foi proposta
- Principal problema: proposta muito ambiciosa. Acabou resultando em uma linguagem muito grande e complexa

Década de 1990

- Influências: gerenciamento de sistemas complexos e, posteriormente, a “Era da Internet”
- Popularização de Orientação a Objetos: tornando-se o paradigma dominante até os dias de hoje
- Linguagens de script: Perl, Javascript, Lua
- Diversas linguagens OO, com destaque à Java

Década de 1990 ⁽²⁾

- Principais Aplicações:
 - Sistemas embarcados (em seu início)
 - Desenvolvimento de aplicações para internet
 - Sistemas Empresariais
 - Programação de rede
 - Aplicativos Android

Java

- Desenvolvida pela empresa Sun Microsystems, sob a coordenação de James Gosling em 1995.
 - Em seu início, puramente OO (simplificou muito o projeto da linguagem)
 - Tem referência, mas não ponteiros
 - Tem Suporte para programação concorrente
 - Usa Garbage Collector

Java (2)

- Máquina Virtual Java:
 - Altera o processo de compilação através de uma máquina virtual que auxilia o processo de tradução
 - Permite que aplicações Java sejam executadas em qualquer local (Sistema Operacional ou Dispositivo)

Década de 2000

- Explosão da Web
- Avanços nas linguagens de script:
 - PHP: uma das primeiras a surgir bastante utilizada na Web 1.0 e 2.0
 - Python e Ruby: aparecem como linguagens de script simples e poderosas

Década de 2000 ⁽²⁾

- Aprimoramento nas linguagens compiladas: C# como alternativa ao Java
- Linguagens híbridas (programação e marcação): JSP, XSLT

Década de 2000 ⁽³⁾

- Novos Domínios:
 - Integração com Bases de Dados.
 - Programação Mobile.
 - Frameworks para desenvolvimento web.
 - Realidade virtual e aumentada
 - Além de evoluções nos domínios já conhecidos.

Década de 2010

- Não é descrita na edição que usamos do livro texto
- Explosão de linguagens, bibliotecas e frameworks
- Amadurecimento das técnicas de programação e boas práticas
- SOLID; Clean Code; Padrões de projeto; Arquiteturas (ex.: MVC); entre outros
 - Veja material complementar

Década de 2010

- Em muitos nichos, alguns trade-offs começam a predominar
 - Simplicidade (facilidade de manutenção) sobre desempenho bruto
 - Preferência por linguagens com tipagem forte e estática (final da década de 2010)
 - Detecção de erros em tempo de compilação

Referências Créditos

- Parcialmente adaptado pelo material gentilmente cedido pelos professores
 - Alessandra Bortoletto Garbelotti Hoffmann (UTFPR – PR)
 - Paulo Ricardo Knob (UTFPR – PR)
 - Matheus Bainy (UTFPR – PR)