



PROGRAMAÇÃO I

INTRODUÇÃO À LINGUAGEM DE PROGRAMAÇÃO JAVA

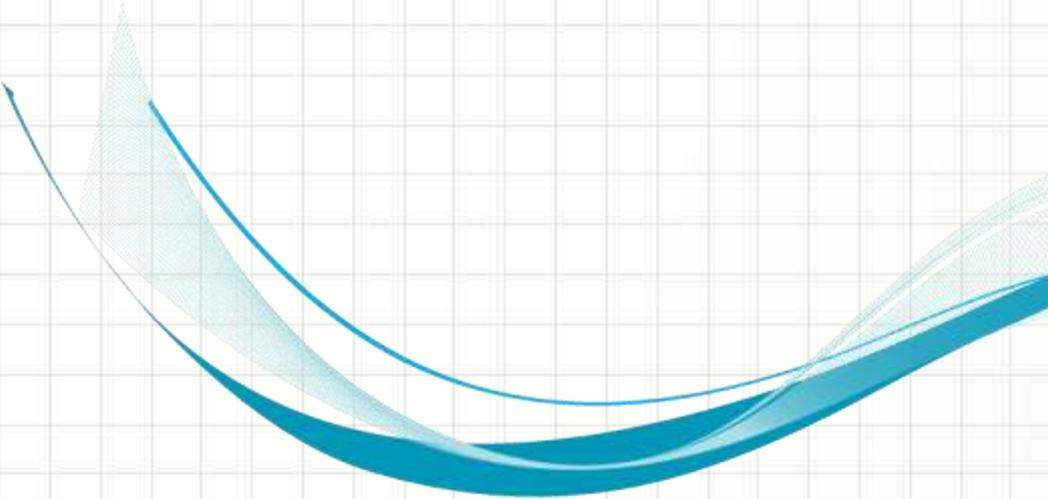
Prof. Dr. Daniel Caetano

2017 - 1

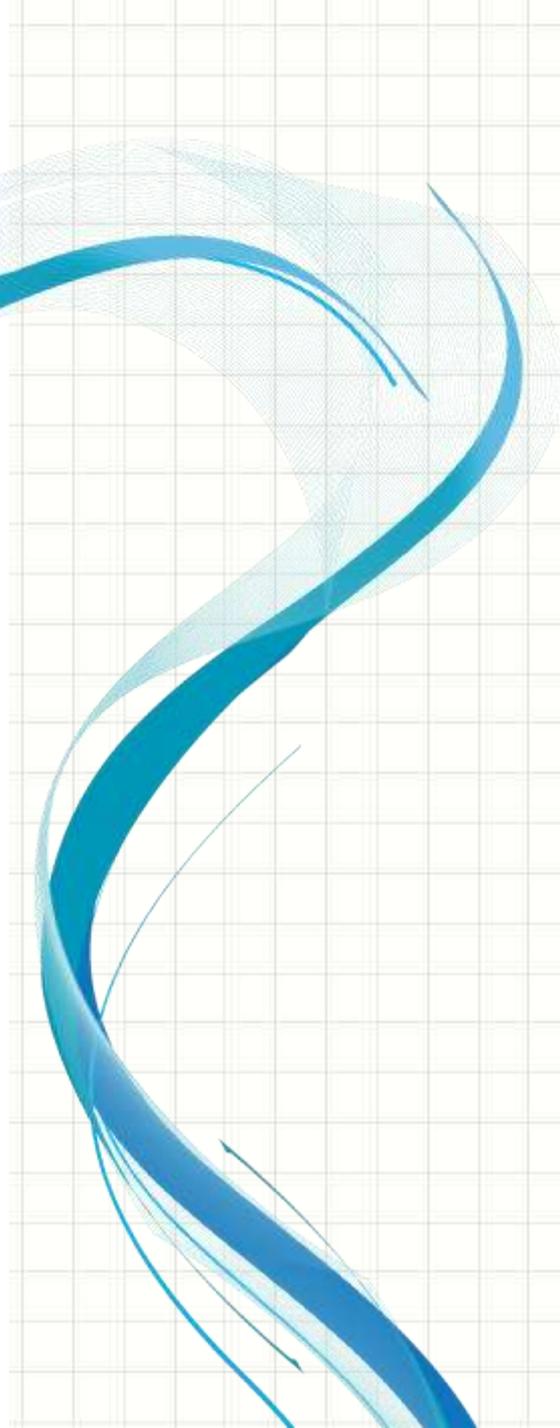
Objetivos

- Conhecer o professor e o curso
- Conhecer as características da linguagem Java
- Conhecer a plataforma Java e o ambiente de programação

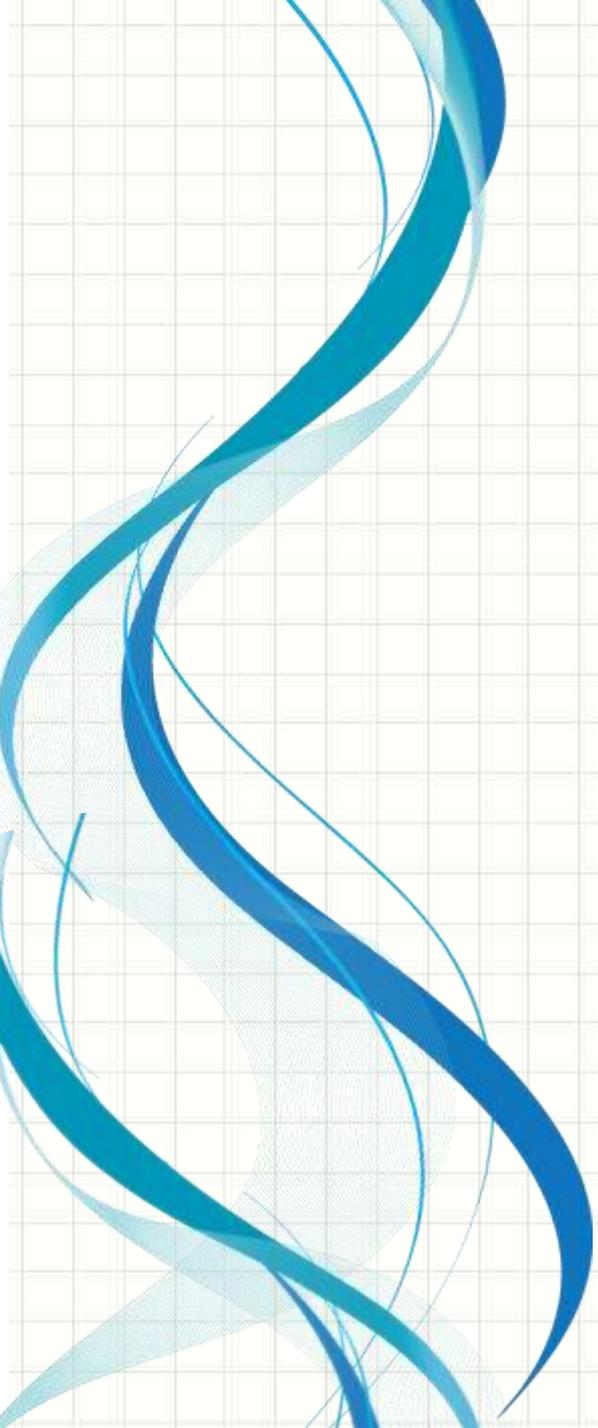




Apresentação



Quem é o
professor?



Quem são os
alunos?



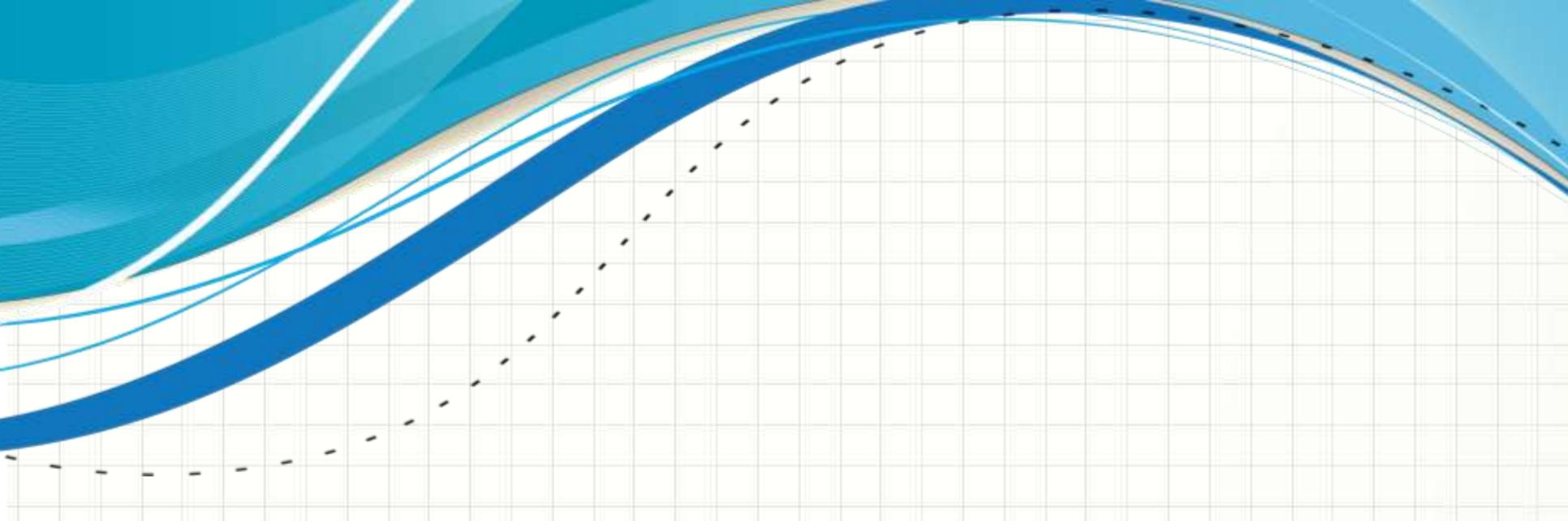
Vamos
começar?

Quem É Quem – Lista de Presença

Professor	Informações de Contato
------------------	-------------------------------

Daniel Caetano	prof@caetano.eng.br
----------------	--

Nome Completo	CPF	Matrícula
Fulano	012.345.678-90	201101123456
Beltrano	012.345.678-91	201101123457
Cicrano	012.345.678-92	201101123458



PLANO DE ENSINO E DE AULA

Plano de Ensino

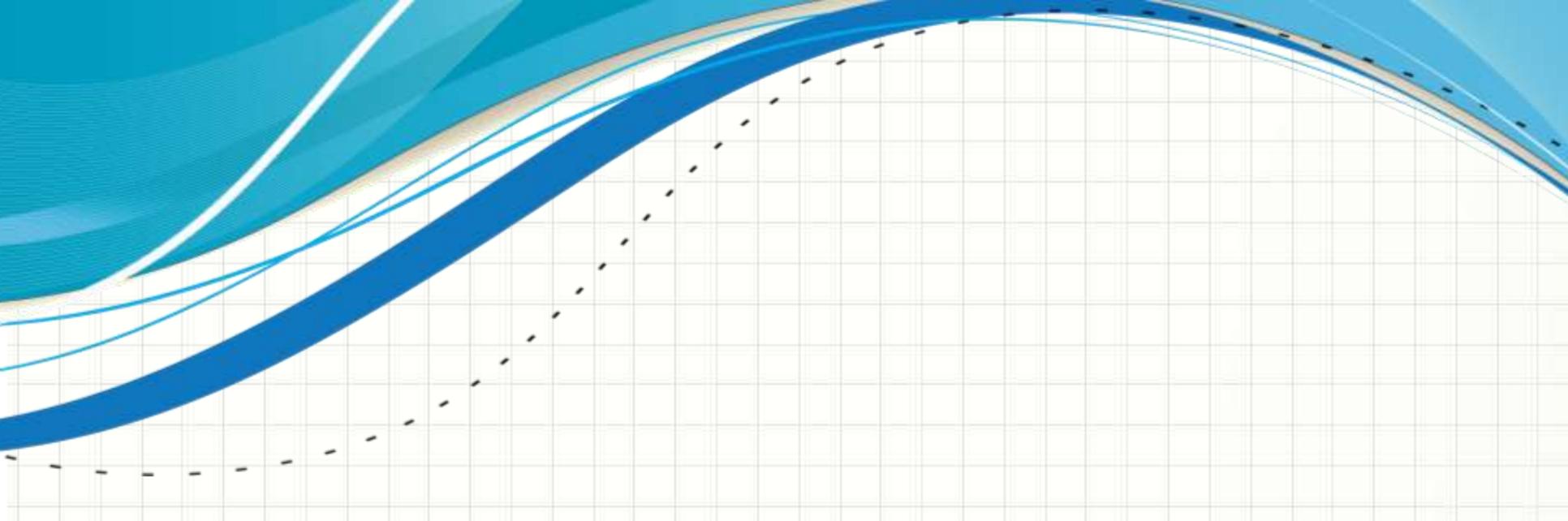
Disponível no SAVA



1. Entre no **SIA**
2. Entre no **SAVA**
3. Clique no **NOME DA DISCIPLINA**
4. Selecione **PLANO DE ENSINO**

Plano de Aula

- 13/02 – 0. Apresentação
- 20/02 – 1. Java Básico
- 06/03 – 2. Java Básico
- 13/03 – Exercícios
- 20/03 – 3. Orientação a Objetos
- 27/03 – 4. Orientação a Objetos
- 03/04 – 5. Orientação a Objetos
- 10/04 – 6. Vetores e Strings
- 17/04 – 7. Vetores e Strings
- 24/04 – **P1**
- **01/05 – FERIADO (Trabalho)**
- 08/05 – 8. Relacionamentos
- 15/05 – 9. Herança e Polimorfismo
- 22/05 – 10. Interfaces
- 29/05 – 11. Exceções
- 05/06 – 12. Coleções
- 12/06 – **P2**
- 19/06 – Vista de Prova
- 26/06 – **P3**



TRABALHOS, DATAS E CRITÉRIO DE APROVAÇÃO

Trabalhos, Datas e Aprovação

Trabalho	Valor	C.H.	Entrega
T1 (Dupla Fixa)	2,0 na AV1	8h	17/04 (SAVA)
P1 (Individual / Com Consulta*)	8,0 na AV1	2h	24/04 (Aula)
T2 (Dupla Fixa)	... na AV2	8h	05/06 (SAVA)
P2 (Individual / Sem Consulta)	10,0 na AV2	2h	12/06 (Aula)
P3 (Individual / Sem Consulta)	10,0 na AV3	2h	26/06 (Aula)

(*) Consulta nos moldes da folha de referência fornecida no site da disciplina.

Trabalhos, Datas e Aprovação

- Atenção ao **prazo de entrega** dos T1 e T2...
- As atividades são grandes, não marque bobeira!
- Não deixe tudo para seu colega!

Trabalhos, Datas e Aprovação – AV1

- Prova P1 à caneta, incluindo o ***preenchimento do cabeçalho***, ganha: 0,25 na P1
- Entregando a folha de consulta (***dentro do padrão***) com a prova, ganha: 0,25 na P1
- A nota do T1 será **somada** à nota da prova P1 para compor a média AV1.

$$\underbrace{AV1}_{0,0 \text{ a } 10,0} = \underbrace{P1}_{0,0 \text{ a } 8,0} + \underbrace{T1}_{0,0 \text{ a } 2,0}$$

Trabalhos, Datas e Aprovação – AV2

- A nota da AV2 é composta **apenas** da nota da avaliação P2! Cuidado! O T2 será apenas para ajustes!

$$\underbrace{AV2}_{0,0 \text{ a } 10,0} = \overbrace{P2}^{0,0 \text{ a } 10,0}$$

Trabalhos, Datas e Aprovação – AV3

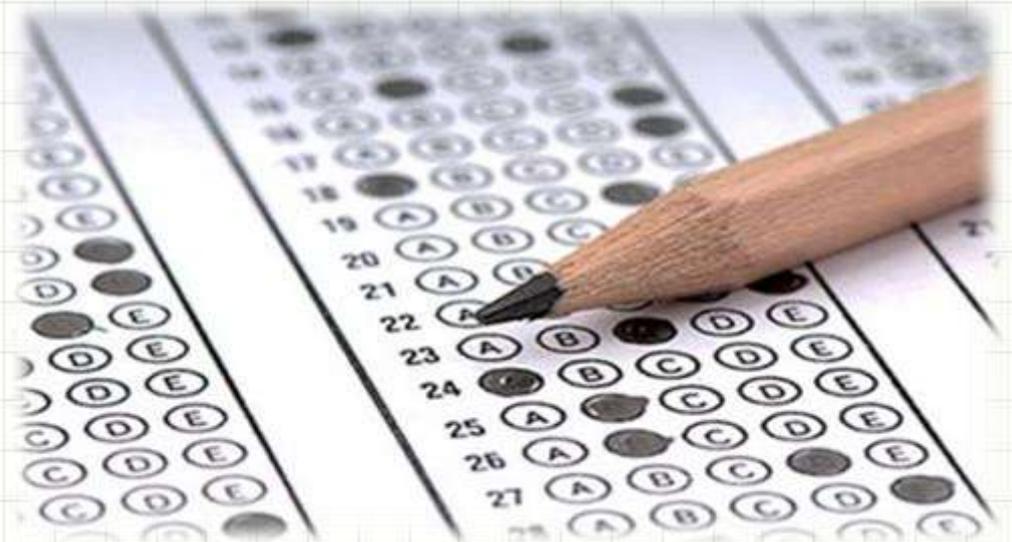
- A nota da AV3 é composta da nota da avaliação P3 somada à nota dos Simulados “Avaliando o Aprendizado”

$$\underbrace{AV3}_{0,0 \text{ a } 10,0} = \underbrace{P3}_{0,0 \text{ a } 10,0} + \underbrace{Sim.}_{0,0 \text{ a } 2,0}$$

Avaliando o Aprendizado

- Quatro Simulados, 5 questões cada
 - Cada questão vale 0,1 na AV3 (se resposta for correta!)
 - Até 2,0 pontos na AV3

<http://simulado.estacio.br/alunos/>



Trabalhos, Datas e Aprovação – Final

A = Maior nota entre { **AV1** , **AV2** , **AV3** }

B = Segunda maior nota entre { **AV1** , **AV2** , **AV3** }

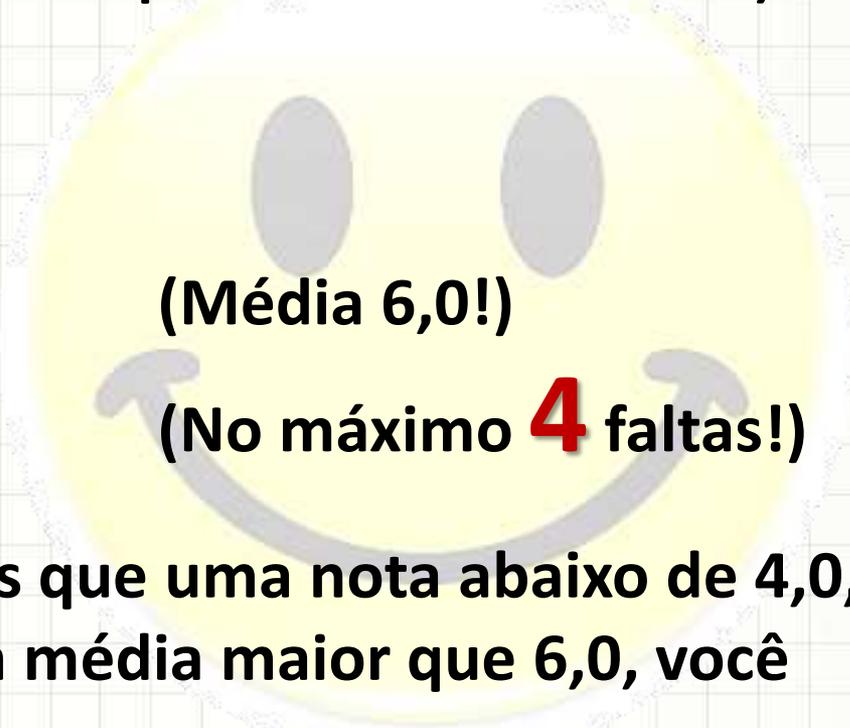
Critérios de Aprovação (TODOS precisam ser atendidos)

1) **A** \geq 4,0

2) **B** \geq 4,0

3) **A** + **B** \geq 12,0

4) Frequência \geq 75%



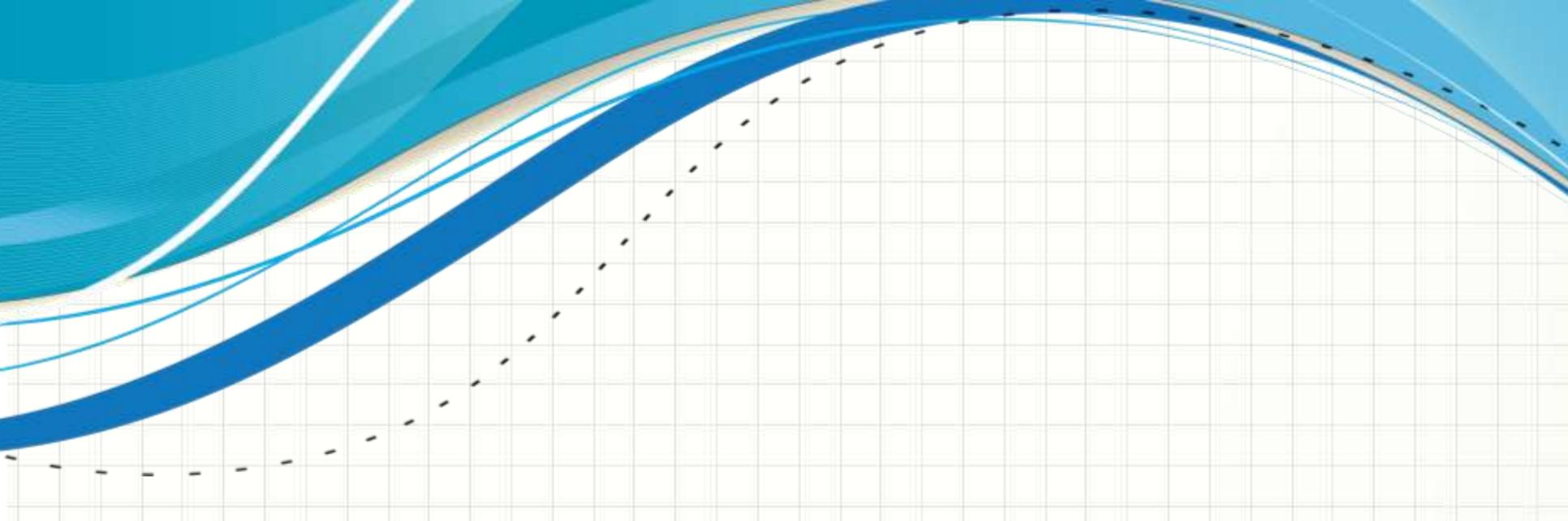
(Média 6,0!)

(No máximo **4** faltas!)

ATENÇÃO: Se você tiver mais que uma nota abaixo de 4,0, ainda que o SIA aponte uma média maior que 6,0, você estará **REPROVADO!**

Relação entre Faltas e Reprovação

- Todos os semestres: alta correlação
 - Mais faltas: piores médias
- Média Presentes / Média Faltantes > 1.5
- AV3 e Reprovações:
 - 4 ou mais faltas: por volta de 90%
 - Menos que 4 faltas: por volta de 50%
 - Menos que 2 faltas: por volta de 20%



BIBLIOGRAFIA E FONTES DE INFORMAÇÃO

Bibliografia

- **Material do Curso**

- Programação I (1ª Edição, 2017)

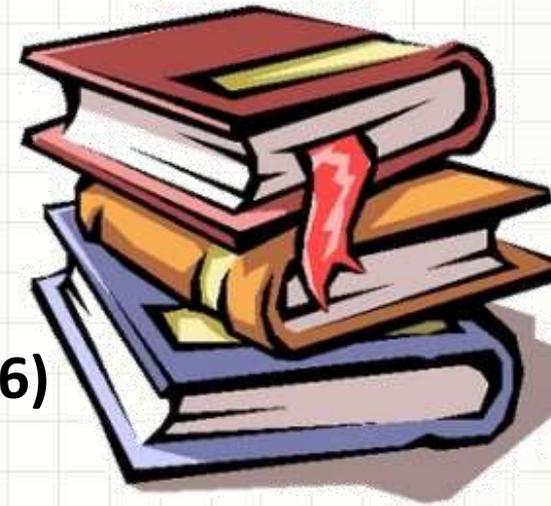
- Fabiano Gonçalves dos Santos
 - Editora Estácio
 - ISBN: 9788555484292 **SAVA!**

- **Biblioteca Virtual**

- Java



Bibliografia



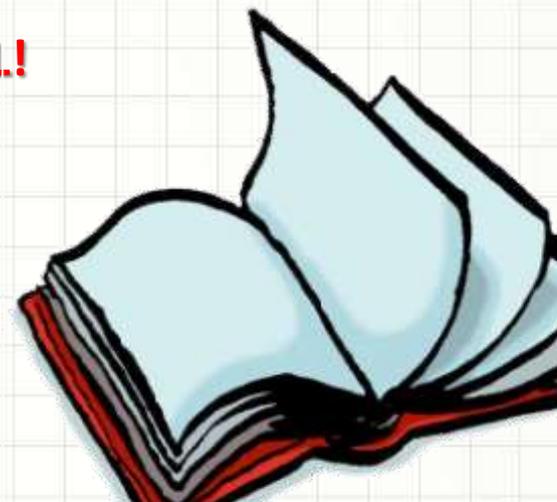
- **Mais Livros!**

- Java: Como Programar+ (10ª Edição, 2016)

- Deitel & Deitel
 - Editora Pearson
 - ISBN: 9788543004792 **BIBLIOTECA VIRTUAL!**

- Core Java – Volume I (8ª Edição, 20010)

- Horstmann & Cornell
 - Editora Pearson
 - ISBN: 9788576053576 **BIBLIOTECA VIRTUAL!**

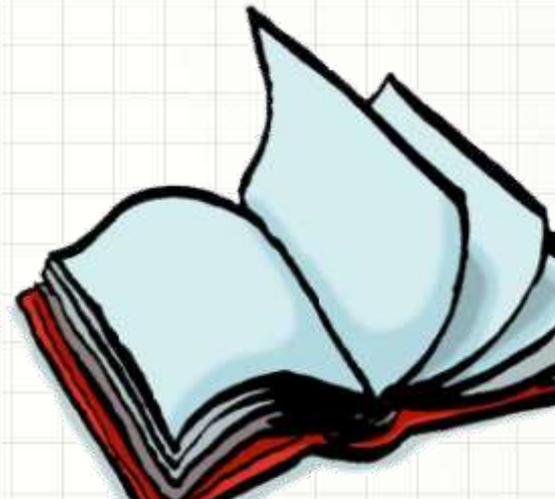
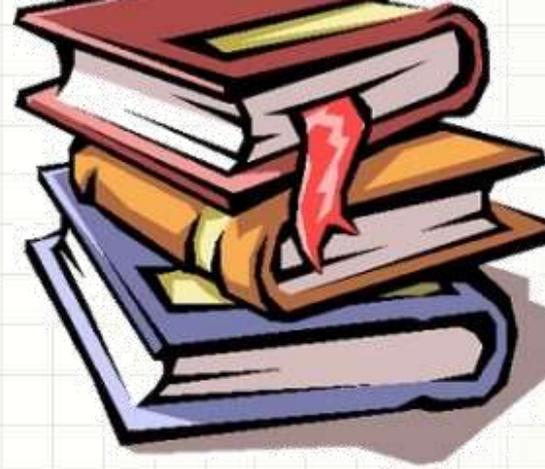


Bibliografia

- Ótimo apoio!

- Prgramação Orientada a Objetos com Java (4ª Edição, 2009)

- Barnes & Kölling
- Pearson Prentice Hall
- ISBN: 9788576051879



Material Didático

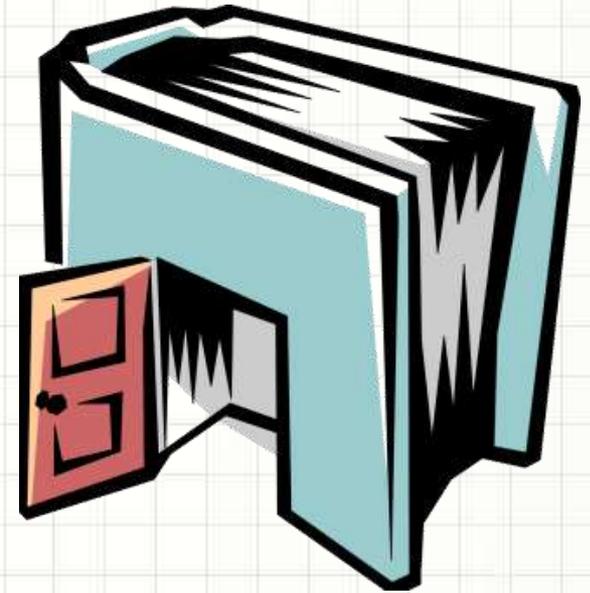
Disponível no SAVA



1. Entre no **SIA**
2. **Entre no SAVA**
3. Escolha a **disciplina**
4. Clique no **ícone** do material didático

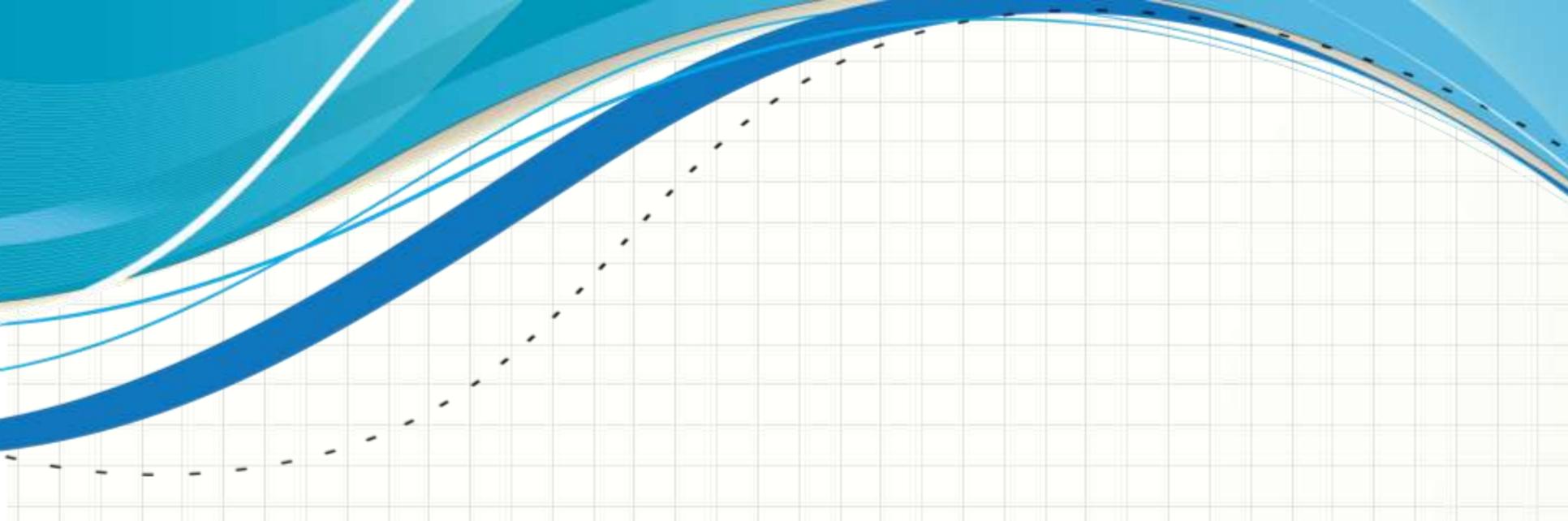
Bibliografia

- **Notas de Aula e Apresentações**



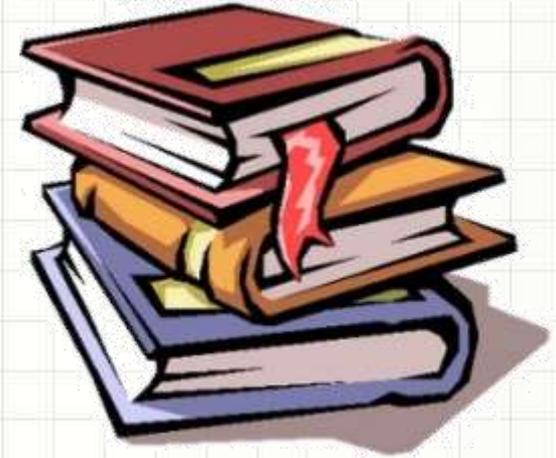
<http://www.caetano.eng.br/>

The screenshot shows the top section of the website. On the left is a photo of Prof. Caetano in a classroom. To the right, the name "Prof. Caetano" is written in a large, elegant script font. In the top right corner, the date and time "17/07/2012, 10:55" and the ID "00021224" are displayed. Below the name are two small flags for Brazil and the United Kingdom. A horizontal navigation bar contains six buttons: "Home", "Ensino", "Pesquisa", "Publicações", "Software", and "Pessoal". The "Ensino" button is highlighted with a red circle. Below the navigation bar, a paragraph of text reads: "Nesta seção você encontra acesso ao material didático desenvolvido pelo Prof. Caetano para os cursos já ministrados. O material está dividido por períodos, visto que boa parte do material não está atualizado."



VOLTANDO À PROGRAMAÇÃO NORMAL:
CONTEXTUALIZAÇÃO

Material de Estudo



Material

Acesso ao Material

Apresentação

<http://www.caetano.eng.br/>
(Programação I – Aula 0)

Material Didático

Programação I – Págs 9 a 17 (não inclui “Tipos de Dados”)

Contextualização

- Algoritmos: resolver problemas
 - Implementar em alguma linguagem
- Linguagens?
 - Início LM & Assembly
 - 195x/6x: FORTRAN, ALGOL 60, COBOL
 - 197x: Pascal, Smalltalk, C, BASIC
 - 198x: C++, Object Pascal / Objective C
 - 1991: VisualBASIC, Oak, Python
 - 1995: PHP, Ruby, Java
 - 2001: C#

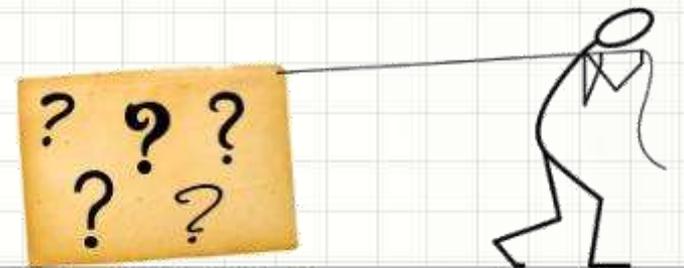
Por quê?



Contextualização

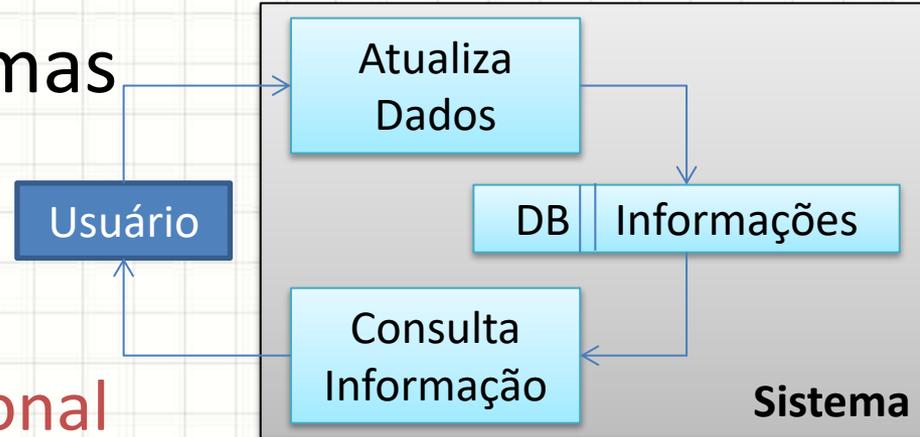
- Problemas Complexos x Custo x Tempo
 - “Sentar e programar” → Projetar
 - Análise e Projeto
- Como implementar um sistema?
 - Compreender o domínio do problema
 - Propor modelo simplificado
 - Propor modelo detalhado
 - Implementar
 - Testar
 - Implantar

Problemas

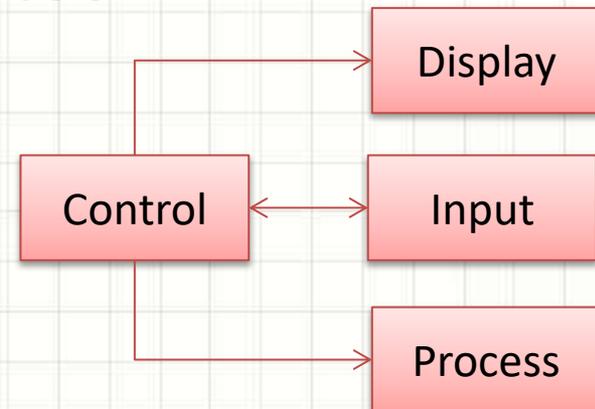
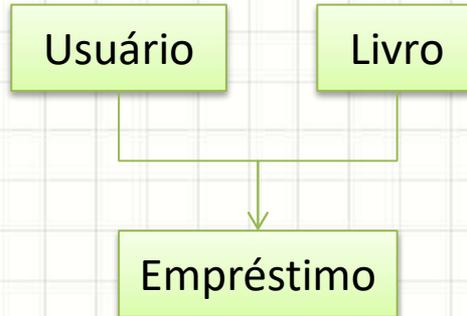


Análise de Sistemas

- LM & ASM: Fluxogramas
- Análise Estruturada
 - Fluxo de Dados (DFD)
 - Decomposição Funcional



- Análise Orientada a Objetos



- O que muda menos ao longo do tempo??

Paradigmas de Programação

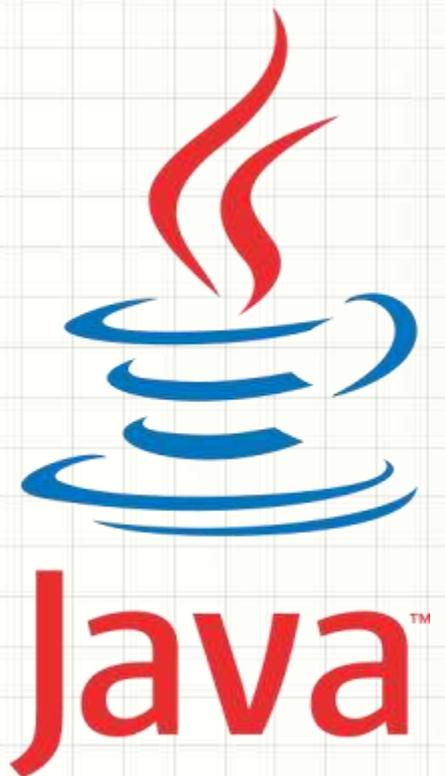
- Linguagens Imperativas
 - Estruturadas/Procedurais
 - Ex.: COBOL, FORTRAN, C, Pascal...
 - Orientada a Objetos
 - Ex.: Smalltalk, C++, Python, Java, C#
- Linguagens Declarativas
 - Funcionais
 - Ex.: Erlang, R, XSLT
 - Lógicas
 - Ex.: Prolog, LISP

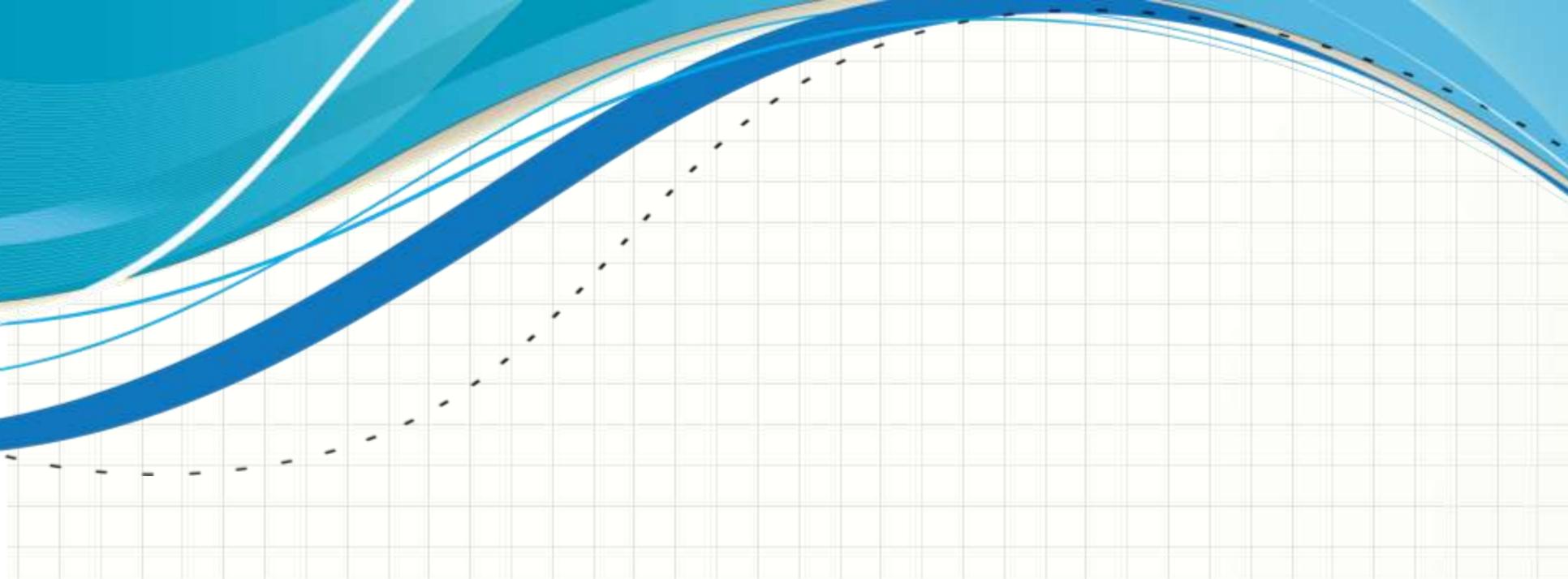


Contexto do Curso

- “Evoluir” de Paradigma:
 - Estruturado → Orientado a Objetos
- Qual linguagem usaremos?
 - JAVA

ORACLE®

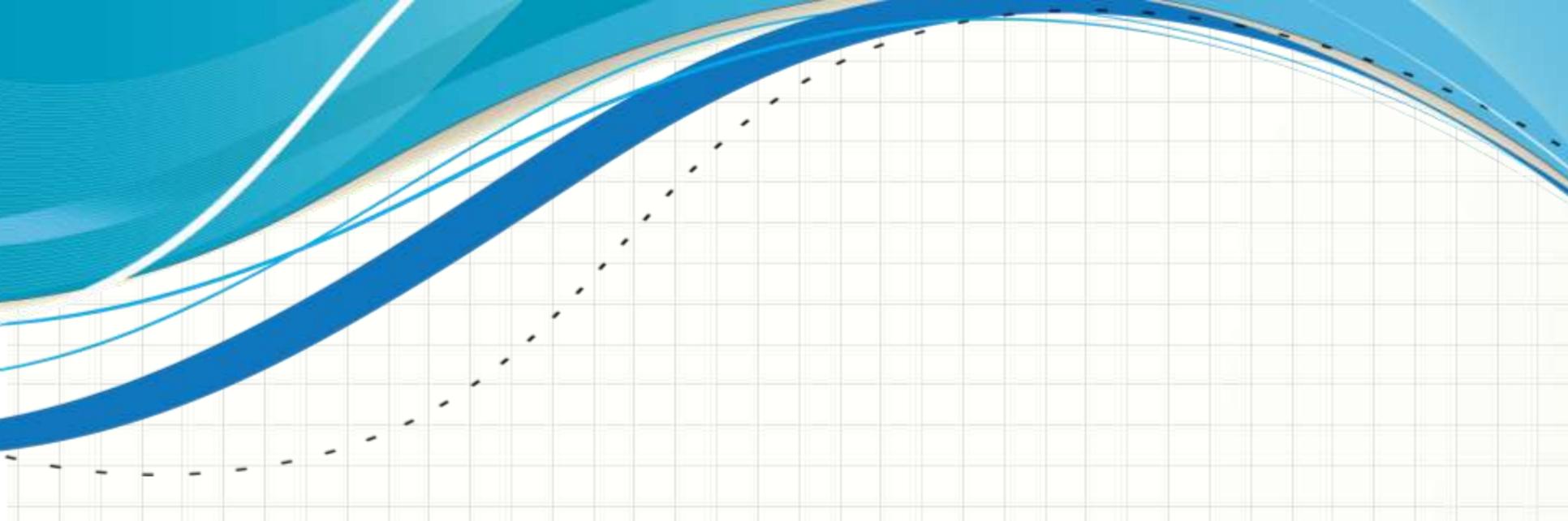




POR QUE JAVA?

Por Que Java

- Linguagem **similar ao C/C++**
- Gerenciamento de exceções e memória
- Vasta biblioteca e suporte Unicode nativo
- Frameworks para desenvolver serviços “out-of-the-box”
- Linguagem mais difundida no mercado de aplicações de grande porte (robustez)
- Alta empregabilidade
- Segurança (sem vírus, criptografia...)
- Paralelismo facilitado



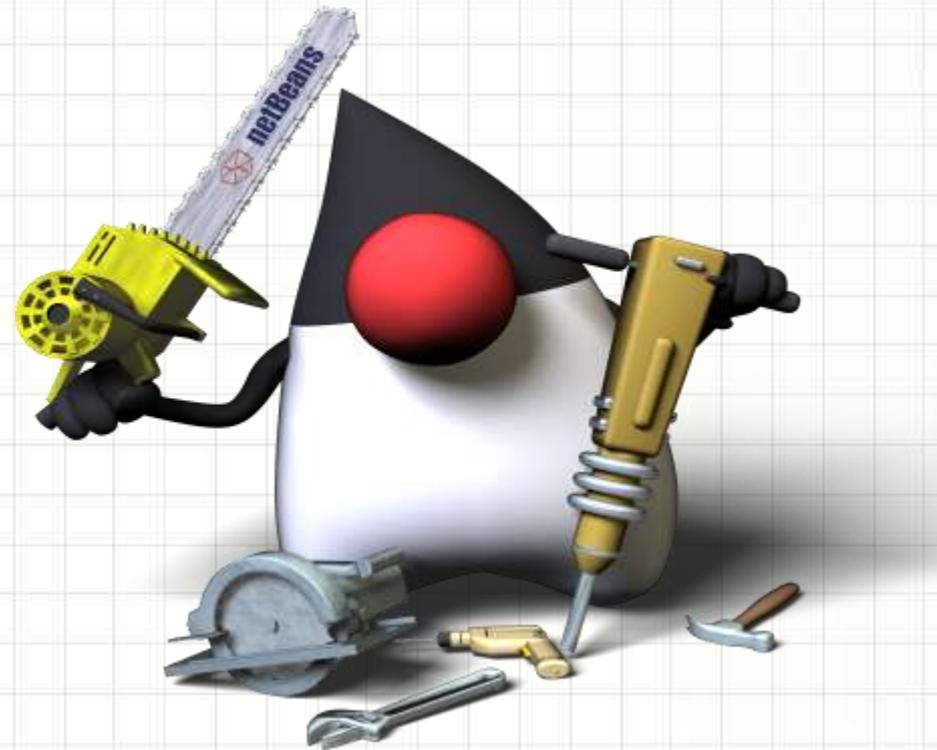
O QUE É JAVA?

O Que é Java

- Uma linguagem **similar ao C/C++** cujo foco é economizar tempo do programador e preservar investimento das empresas
- Como?
 - Uma vasta biblioteca pronta
 - Exigência de atendimento aos padrões mais modernos de desenvolvimento de software
 - Compatibilidade quase universal: um programa pode ser criado de maneira a ser executado em praticamente qualquer equipamento que seja capaz de processamento

O Que é Java

- Como isso se tornou realidade?
- É importante conhecer a origem do Java...
- E as ideias que fundamentam seu funcionamento



Um Pouquinho de História

- Até 1994: Foco em Dispositivos Embarcados
- 1995~: Implementação em navegadores Web
- Atualmente: Plataforma ampla com múltiplas soluções
- Oak / Java – Sem sucesso inicial
- Applets – Ação na Internet
- Java SE, EE e ME – Difusão

Como Funciona o Java ?

- Como um programa pode rodar em qualquer lugar?
- Vejamos como um programa “normal” é gerado e executado...
- E depois comparemos com a forma com que um programa Java é gerado é executado!

Como Funciona o Java ?

- Linguagem Tradicional



Programador



```
#include <io...  
int main(void)  
{  
  cout << "Oi";  
}
```

Código Fonte



Compilador



```
001010101010  
101010101010  
110111011011  
111110010101
```

Código Binário
de PC



Computador PC

Como Funciona o Java ?

- Linguagem Tradicional



Programador



```
#include <io...  
int main(void)  
{  
  cout << "Oi";  
}
```

Código Fonte



Compilador



```
001010101010  
101010101010  
110111011011  
111110010101
```

Código Binário
de PC



Celular

Como Funciona o Java ?

- Linguagem Java



Programador



```
import java...  
int main(Stri...  
{  
  System.out...
```

Código Fonte



Compilador



```
001010101010  
101010101010  
110111011011  
111110010101
```

Byte Code Java



Computador PC

Como Funciona o Java ?

- Linguagem Java



Programador



```
import java...
int main(Stri...
{
System.out...
```

Código Fonte

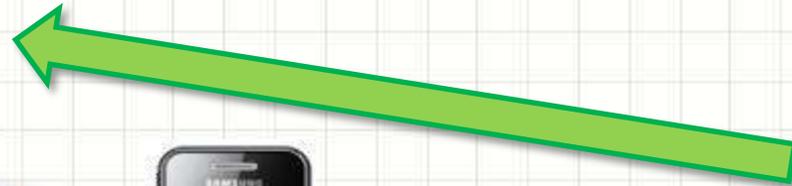


Compilador

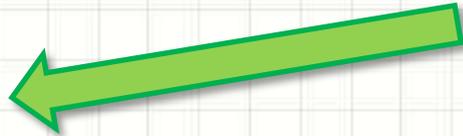


```
001010101010
101010101010
110111011011
111110010101
```

Byte Code Java



Computador PC com Interpretador Java



Celular com Interpretador Java

O Que é o Interpretador Java?

- O programa Java, depois de compilado, fica em um formato incompreensível para o computador, chamado **Byte Code**.
- O Interpretador Java é um programa que lê o Byte Code e traduz para o computador, indicando o que ele precisa fazer.
- Basicamente, o interpretador Java faz o trabalho de um **intérprete**.
- Cada equipamento precisa ter seu próprio interpretador Java.

Nomenclatura do Java

- O nome oficial do Interpretador Java é “**Java Virtual Machine**”, carinhosamente apelidado de **JVM**.
- Para executar aplicações Java, é necessário instalar a JVM.
- A JVM sozinha, porém, não contém as **bibliotecas** necessárias para executar tudo que um programa Java precisa.
- Assim, a Oracle distribui um pacote chamado **Java Runtime Environment (JRE)** contendo a JVM e as bibliotecas oficiais do Java.

Nomenclatura do Java

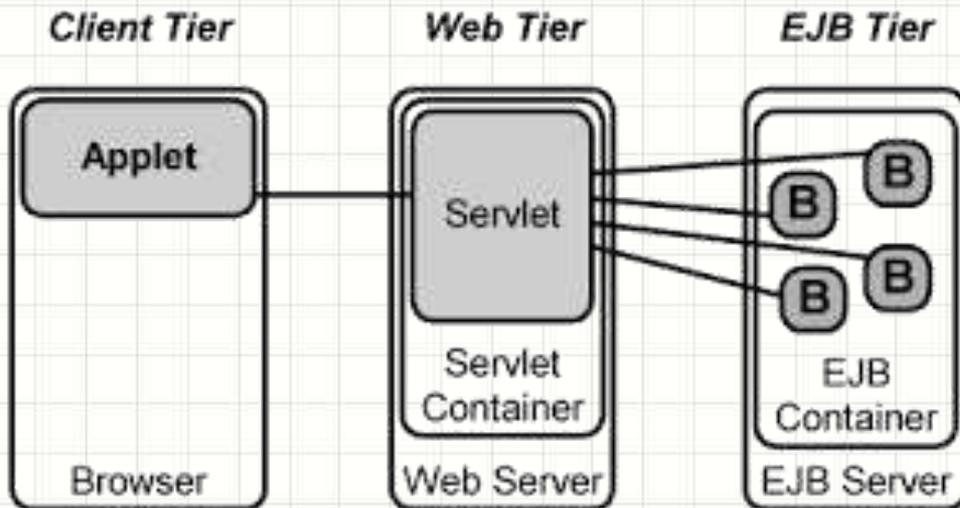
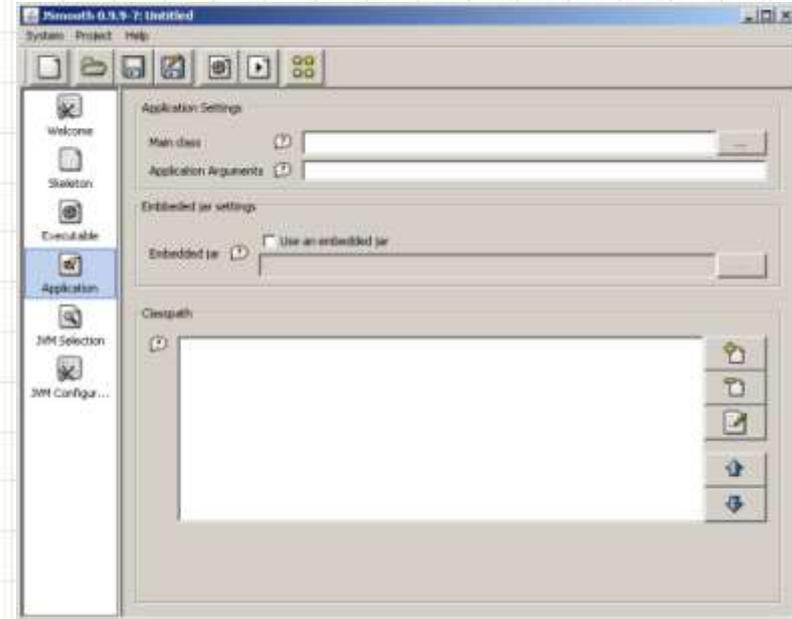
- O JRE inclui apenas os pacotes necessários à execução de programas Java.
- Para poder **gerar** programas Java, é necessário baixar um pacote mais completo, chamado **Java Development Kit (JDK)**.
- O JDK inclui, além dos elementos do JRE, também as ferramentas de desenvolvimento do ambiente Java.

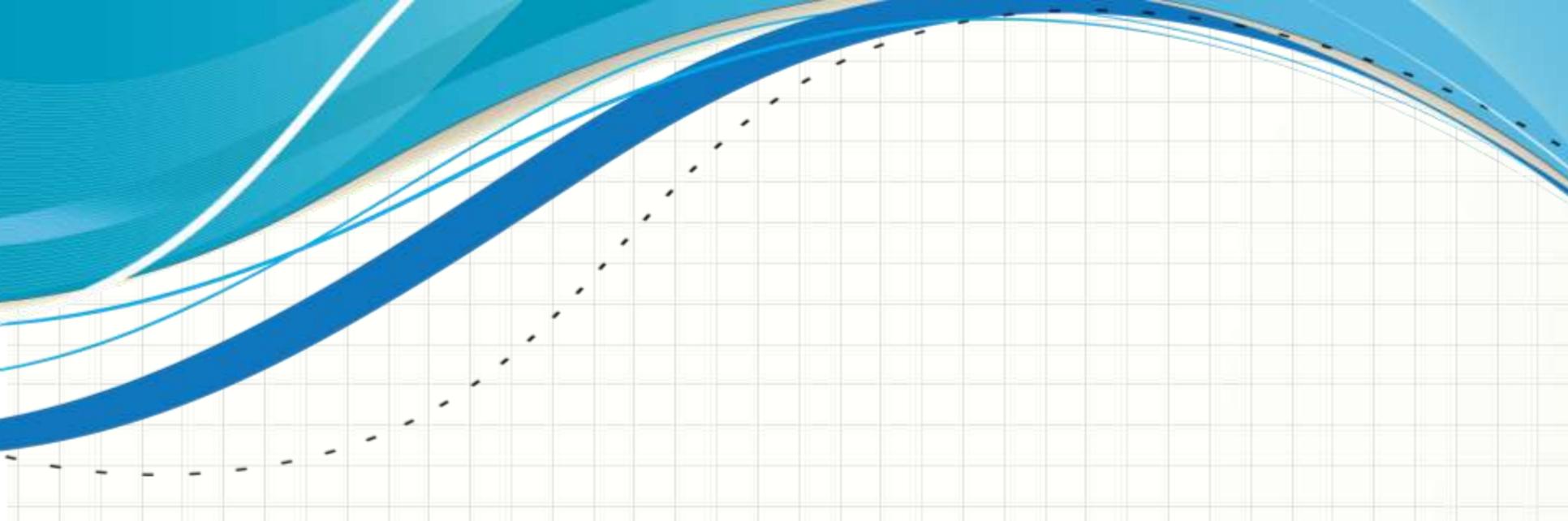
Nomenclatura do Java

- Como o JRE e o JDK se tornaram muito grandes, a Oracle organizou três pacotes diferentes, de acordo com as necessidades de cada programador:
- Java Micro Edition (Java ME): para aplicações portáteis, isto é, aquelas que executam no celular do usuário.
- Java Enterprise Edition (Java EE): para aplicações que executam em um servidor, isto é, respondendo requisições.
- Java Standard Edition (Java SE): para aplicações desktop, isto é, aquelas que executam no computador do usuário.
- Neste curso usaremos o Java SE, já que desenvolveremos aplicativos desktop.

Tipos de Aplicações Java

- Aplicações tradicionais
- Java Applets
- Java Servlets





PERGUNTAS?