



# Aula 1

---

## Programação em Java

### Introdução

Programação em Java 2006-2007



## Objectivos

---

Estudo de técnicas que conduzam ao uso eficaz do computador, como uma ferramenta auxiliar na resolução de problemas

### O que impõe:

- compreender os conceitos básicos sobre programação que permitam adquirir competências ao nível do desenvolvimento de programas de **pequena e média** complexidade.
- identificar áreas de aplicação e limitações dos computadores na resolução de problemas.
- saber especificar de forma precisa os problemas a resolver e construir descrições detalhadas e rigorosas de soluções que possam ser implementadas usando os computadores.
- implementar essas soluções em linguagem Java
- usar um ambiente de desenvolvimento, onde os programas possam ser escritos, documentados, testados e validados.

Programação em Java 2006-2007

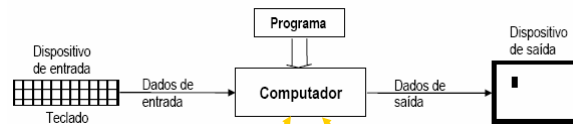
2

# O que é um computador?

Uma máquina programável capaz de efectuar qualquer tipo de processamento automático de informação.

**Computador = Hardware (parte física) + Software (programas e dados)**  
*Elementos mais importantes de hardware:*

- **memória central** (RAM - *Random Acces Memory*): armazena programas e dados temporários
- **unidade processadora central** (CPU – *Central Process Unit*): executa instruções que são reconhecidas por esse tipo de CPU



O computador executa programas sobre dados comunicados do exterior por um **dispositivo de entrada**

Logo encaminha os resultados desse processamento, também para o exterior, através de um **dispositivo de saída**;

Programação em Java 2006-2007

3

# O que é um programa?

Uma seqüência de operações elementares (**instruções**) que pode ser executada por um **computador**

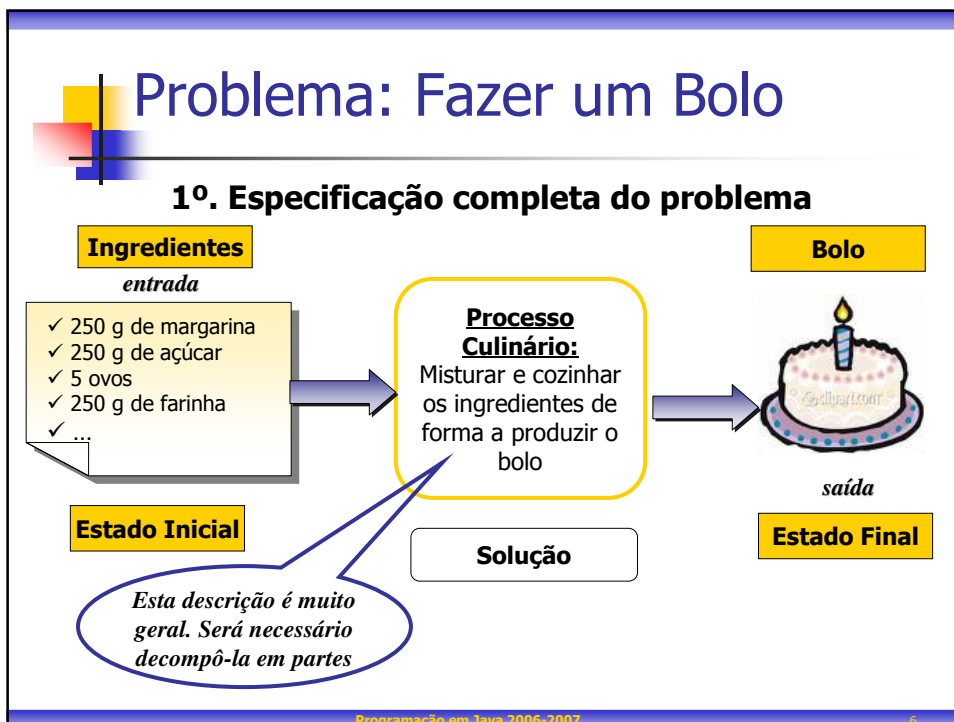
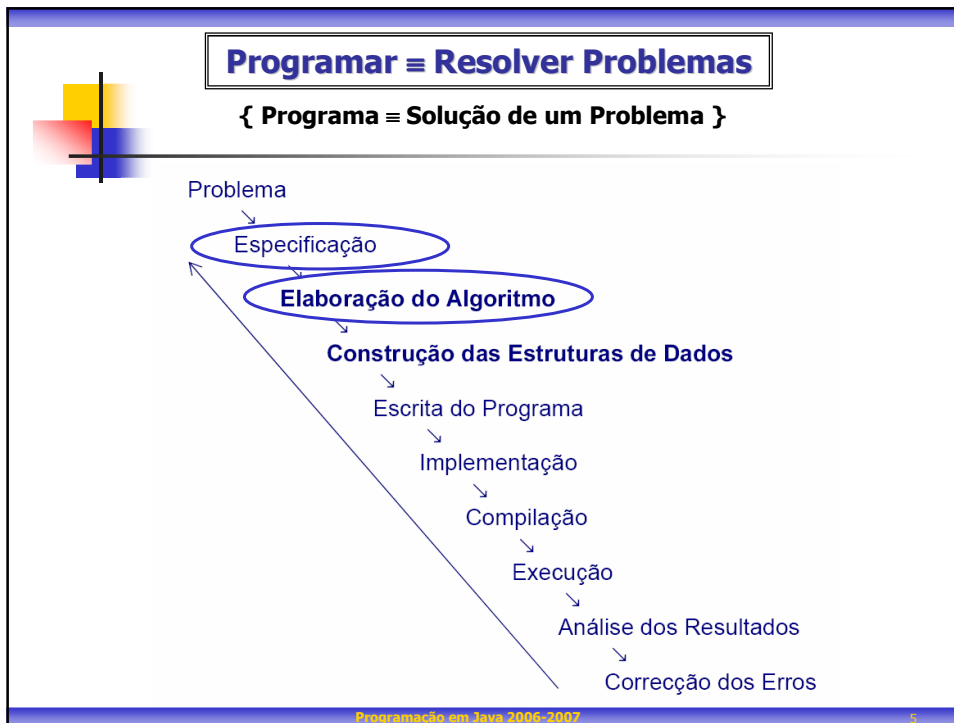
**Programa = Algoritmos + Estruturas de Dados**

*Um bom programa impõe:*

- **Clareza:** O programa deve reflectir claramente a estrutura do algoritmo. Deve ser fácil de **ler**, corrigir, ampliar ou modificar, mesmo por outro programador
- **Correcção:** O programa deve cumprir exactamente as especificações.
- **Eficiência:** O programa deve tentar minimizar, tanto o seu **tempo** de execução, como o **espaço de memória utilizado**.

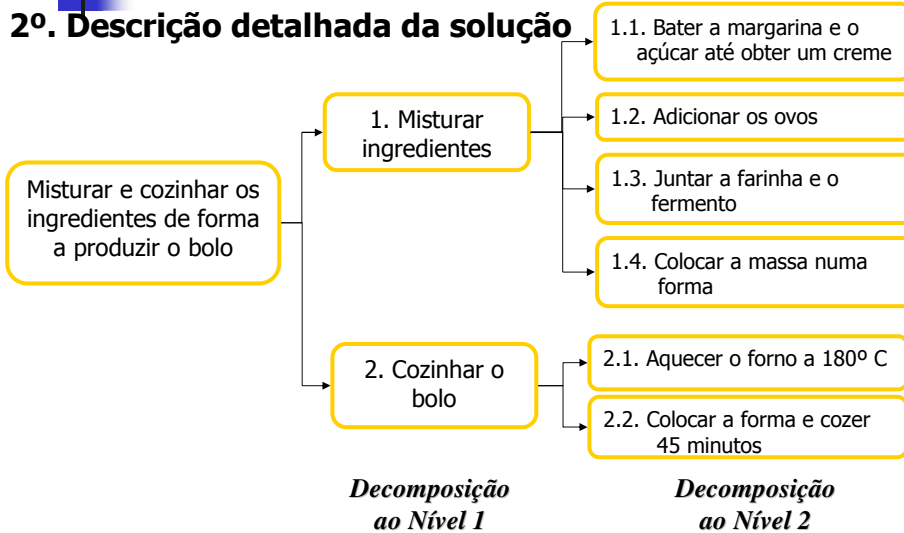
Programação em Java 2006-2007

4



# Problema: Fazer um Bolo

## 2º. Descrição detalhada da solução



Programação em Java 2006-2007

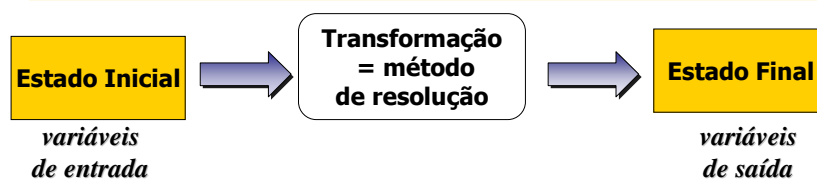
# Especificação do Problema

*Tipos de problemas que o computador resolve:*

- problemas **completamente especificados**
- problemas que se podem **decompor** numa combinação de um número finito de problemas completamente especificados.

Um problema **completamente especificado** é aquele em que:

- as **variáveis de entrada** e de **saída** estão **perfeitamente identificadas**
- Se conhece uma solução, ou seja, um método que permite obter, **de forma unívoca**, os valores das variáveis de saída em função dos valores das variáveis de entrada.



Programação em Java 2006-2007



## Especificação Completa do Problema

### Considerações a ter em conta:

- Considerar a resolução de **classes de problemas** e não de problemas particulares, o que implica uma **descrição algébrica** da solução do problema
- Estabelecer claramente a gama de valores permitida para as variáveis de entrada
- A solução descrita deve contemplar alternativas para toda a gama de valores das variáveis de entrada, eliminando toda e qualquer ambiguidade.



## O que é um algoritmo?

Uma sequência não ambígua de instruções (*operações sobre os dados*) que num número finito de passos ou produz os resultados de um problema ou indica que a solução não pode ser obtida

- Um algoritmo é uma descrição **detalhada e rigorosa** da solução de um problema. Uma maneira de escrever algoritmos é usando **pseudo código**:
  - escrita abreviada, que pode ser baseada em **português**
  - cada uma das operações (passo do algoritmo) tem que ser enunciada sem ambiguidade, numa forma clara e concisa

## Regras para descrever algoritmos

A transcrição do pseudo código para uma linguagem de programação dá origem ao **programa**

- Para se garantir uma passagem natural do pseudo código ao programa:
  - as operações descritas devem reproduzir às operações que o computador seja capaz de executar
    - Receber e fornecer informação
    - Guardar informação para posterior utilização
    - Operações Aritméticas
    - Operações Lógicas
    - Instruções decisórias: selecção (escolher entre variáveis)
    - Instruções repetitivas: repetir um grupo de acções
  - as regras gramaticais do pseudo código devem ser semelhantes às da linguagem de programação que se vai utilizar.

## Decomposição da solução

- **Nível 1** (*modelo básico*):
  1. Leitura dos valores das variáveis de entrada
  2. Processamento
  3. Escrita dos valores das variáveis de saída



*As três etapas correspondem directamente à descrição funcional do que é um computador, e de como ele executa um programa.*

- **Nível 2** (*modelo básico refinado*):
  - As etapas do modelo básico podem a sua vez ser decompostas em sub seguintes sub-etapas.

## Exemplo 1: Conversão de distâncias (milhas para quilômetros)

**Formulação do problema:** ler uma distância expressa em milhas a partir do teclado, convertê-la para quilômetros e apresentar o resultado no ecrã.

**Variável de entrada:** **MILHAS** (distância expressa em milhas)  
valor numérico positivo ou nulo

**Variável de saída:** **KM** (distância expressa em quilômetros)  
valor numérico positivo ou nulo

**Solução:** **KM** = 1.609 x **MILHAS**

## Exemplo 1: Conversão de distâncias

### Algoritmo

(decomposição ao nível 1)

**Nome:** Conversão de distâncias em milhas para km

{

**Leitura e validação** de uma distância em milhas (**MILHAS**);

**Conversão** da distância de milhas para km (**MILHAS, KM**);

**Impressão** no ecrã da distância em km (**KM**);

}

## Exemplo 1: Conversão de distâncias

### Algoritmo

(decomposição ao nível 2)

*leitura* → **Nome:** Leitura e validação de uma distância em milhas (**MILHAS**)

```
{
  repetir {
    impressão no ecrã da mensagem "Distância em milhas?";
    leitura do valor a partir do teclado (MILHAS);
  } enquanto MILHAS < 0;
}
```

*processamento* → **Nome:** Conversão da distância de milhas para km (**MILHAS, KM**)

```
{
  KM = 1.609 x MILHAS;
}
```

*escrita* → **Nome:** Impressão no ecrã da distância em kms (**KM**)

```
{
  escrita no ecrã da mensagem "Distância em kms= ";
  escrita no ecrã do valor de KM;
  mudança de linha
}
```

Programação em Java 2006-2007

15

## Regras para descompor a solução

- descrição 'top-down'
  - controlo do grau de abstracção de cada nível
  - limitação do seu número de operações
- refinamento progressivo da descrição
  - passagem gradual de uma descrição em linguagem natural para uma descrição algébrica mais detalhada
    - faz-se, à medida das necessidades, a definição das variáveis internas, a clarificação das expressões booleanas decisórias ou de terminação de ciclo repetitivo, etc;
  - especificação formal de novas operações
  - caracterização rigorosa das dependências de informação.

Programação em Java 2006-2007

16



## Bibliografia

---

- António José Mendes, Maria José Marcelino. Fundamentos de programação em JAVA 2.FCA – Editora de informática, 2003.
- Introdução às aulas de Análise e desenvolvimento de Algoritmos. Apontamentos da professora Rosália Rodrigues  
(on-line em <http://www2.mat.ua.pt/rosalia/cadeiras/ADAant4/Introducao.pdf>)
- Apontamentos das aulas TP da disciplina Programação 1 (2006-2007). Departamento de Electronica, Telecomunicações e Informática. Universidade de Aveiro