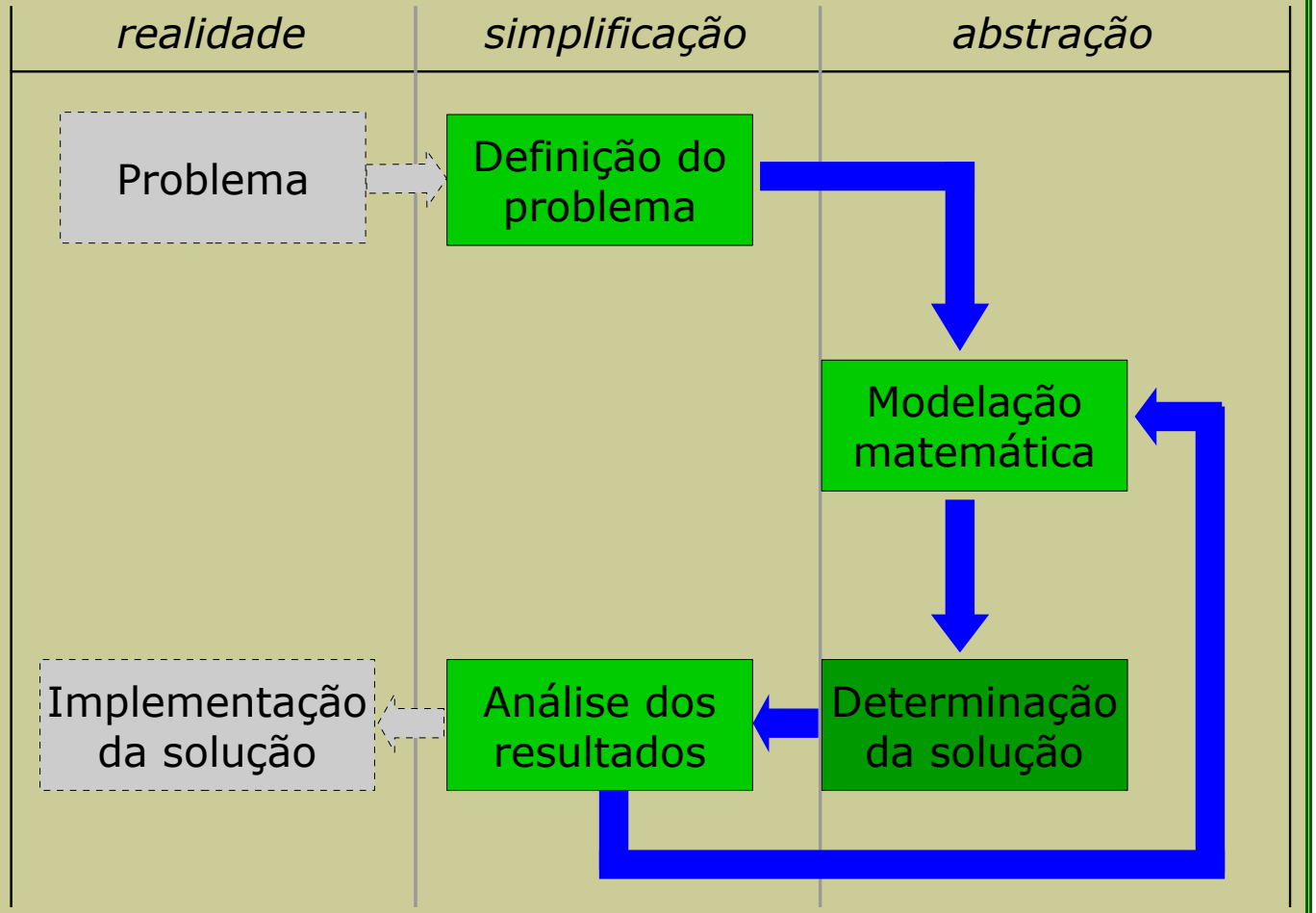
Algoritmos

Resolução de um problema real

A solução de um problema real pode ser obtida em 4 etapas

- Definição do problema
- Modelação matemática
  (construção do modelo)
- Determinação da solução
- Análise dos resultados
  - validação do modelo
  - análise da solução

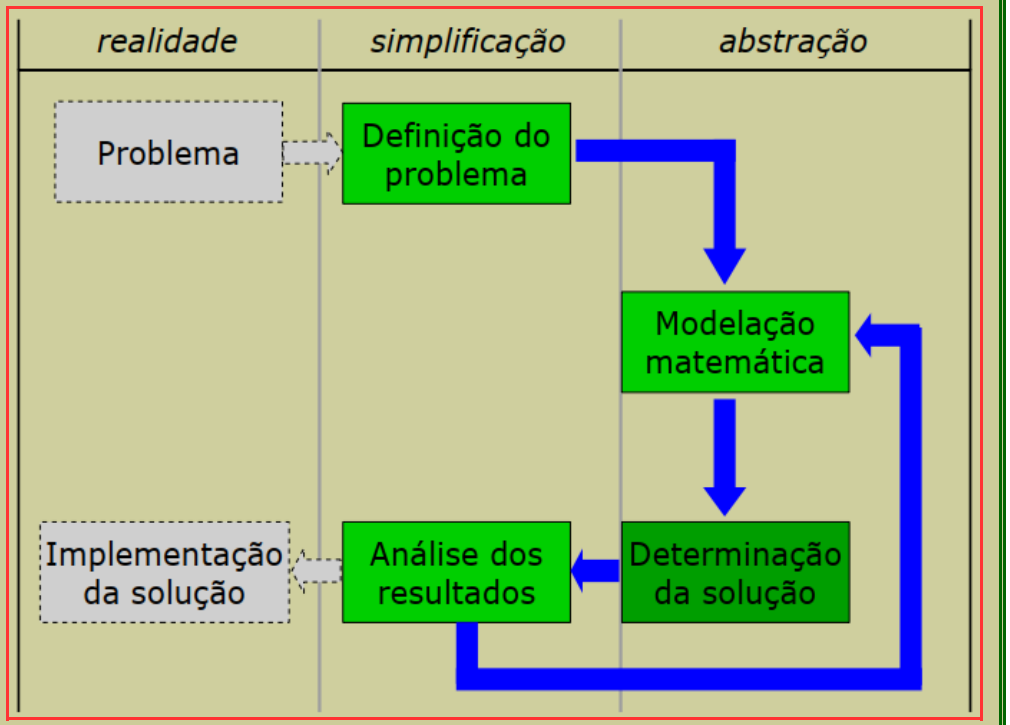


Definição do problema

- Definir/descrever o problema real que se pretende resolver

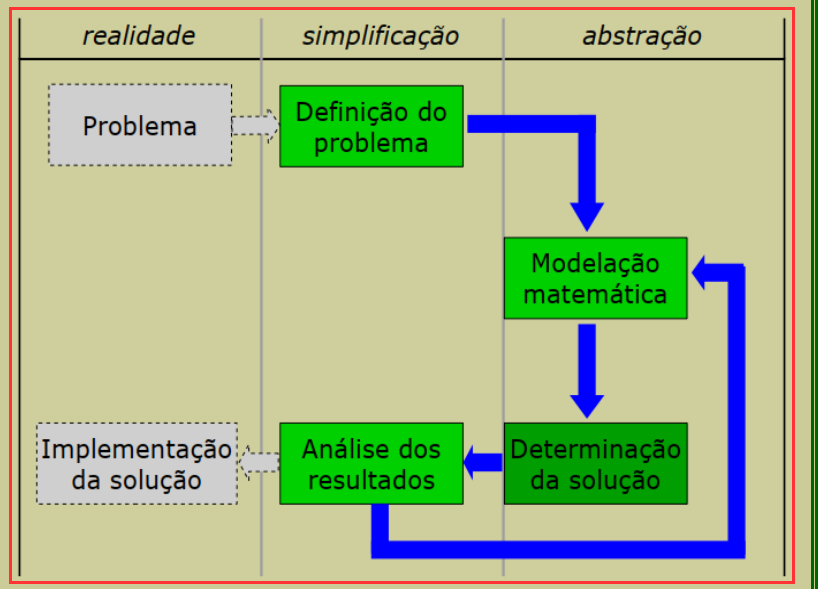
Modelação matemática (construção do modelo)

- O problema real é transformado num problema matemático, através de uma formulação matemática denominada **modelo matemático**
- Geralmente, o modelo matemático possui mais soluções que o problema real



Determinação da solução

- Etapa composta pelas seguintes fases:
  - **escolha do método** matemático mais apropriado para resolver o modelo matemático obtido na modelação matemática
  - **elaboração do algoritmo**, descrevendo o método matemático escolhido antes
  - **implementação do algoritmo**, traduzindo o algoritmo usando uma linguagem de programação de computadores (codificação do programa)



  - **obtenção da solução**, através da execução do programa
- A escolha do método mais apropriado deve considerar os seguintes aspetos:
  - precisão desejada para os resultados
  - capacidade do método em conduzir aos resultados desejados
  - **esforço computacional despendido** (tempo de processamento e memória)

Determinação da solução

- **Elaboração do algoritmo**
  - a descrição do algoritmo é feita através de um **conjunto de comandos** que, quanto ativados (executados), resultam numa sucessão finita de ações (acontecimentos)
  - em vez de se implementar um método diretamente numa linguagem de programação, é preferível descrevê-lo através de uma **notação algorítmica**
    - é possível abstrair-se dos detalhes da linguagem de programação do computador e concentrar-se apenas nos aspetos matemáticos do método (raciocínio matemático)
    - melhora o entendimento do algoritmo, facilitando assim a implementação do método em qualquer linguagem de programação

Determinação da solução

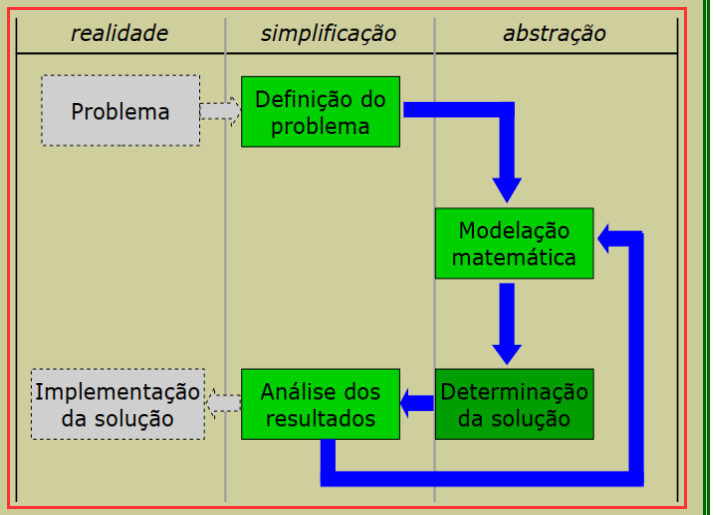
- Implementação do algoritmo (codificação do programa)

- o algoritmo é implementado na linguagem de programação escolhida
- uma vez que os aspetos matemáticos do método já foram pensados na fase de elaboração do algoritmo, agora apenas é necessário preocupar-se com os detalhes de implementação na linguagem adotada

- Obtenção da solução (execução do programa)

- o código obtido da implementação do algoritmo deve ser executado pelo computador
- se for detetado algum erro lógico na fase de processamento (a execução do programa produzir resultados inesperados), então deve-se regressar à fase de elaboração do algoritmo para o corrigir
- se não for detetado qualquer erro lógico na fase de processamento, então foram obtidos os resultados do modelo matemático (a solução)

Análise dos resultados



- **Verificar a consistência da solução obtida**
  - para o modelo matemático (validação do modelo)
  - na sua adequação ao *problema real* (análise da solução)
- **Validação do modelo**
  - se a solução não for satisfatória para o modelo (modelo não válido), então deve-se
    - construir um novo *modelo matemático* (uma nova formulação matemática)
    - determinar as soluções do novo modelo matemático
- **Análise da solução**
  - alguns modelos matemáticos podem
    - produzir várias soluções (e não apenas uma)
    - algumas delas (ou todas) não terem sentido (ex: tempo negativo, ...)
  - um dos objetivos desta etapa é identificar qual a solução válida para o problema real entre as várias obtidas para o modelo matemático (se existirem)

Elaboração de Algoritmos

Algoritmo, Estrutura de dados e Programa

- Um **algoritmo** é uma descrição detalhada (passo a passo) do método escolhido para determinar a solução do modelo matemático que representa o problema real
  - o problema real tem dados que têm que ser tratados computacionalmente
  - o modelo matemático que foi construído para representar matematicamente o problema real, tem que definir os dados e os resultados do problema real matematicamente
  - o método que foi criado para resolver o modelo matemático, é definido através de expressões aritméticas e lógicas para tratar os dados e os resultados (solução)

- Um **programa** é a tradução de um algoritmo numa determinada linguagem de programação, de modo que possa ser executado pelo computador
  - um **programa** é a expressão de um **algoritmo**

- Uma **estrutura de dados** é a forma ou processo como os dados são organizados e tratados (acedidos e alterados)

Algoritmo, Estrutura de dados e Programa

- Um programa pode ser definido da seguinte forma

  Programa = Estrutura de Dados + Algoritmo

  - as estruturas de dados tratam os dados
  - o algoritmo manipula os dados

- Processo de organização dos dados pode determinar a eficiência do algoritmo
  - algoritmos simples podem requerer estruturas de dados complexas
  - algoritmos complexos podem requerer estruturas de dados simples

- Para maximizar a eficiência do algoritmo
  - definir as estruturas de dados a usar em simultâneo com o desenvolvimento do algoritmo

- Os algoritmos eficientes são aqueles que usam “boas” estruturas de dados

Passos na elaboração de um algoritmo

- A elaboração de algoritmos é uma das **fases mais importantes** na resolução de um problema real, pois traduz o método escolhido para resolver o modelo matemático construído, numa forma próxima da linguagem de programação
- Passos na elaboração do algoritmo
  - identificar os dados de entrada
  - identificar os dados de saída (resultados)
  - identificar o que é preciso para transformar os dados de entrada nos resultados (método dividir-para-conquistar)
  - construir o algoritmo
  - testar manualmente o algoritmo
  - executar o algoritmo

Método Cartesiano de Dividir-para-Conquistar

- Também conhecido por **método descendente** ("*top-down method*") ou **método de refinamento passo-a-passo**
- Consiste em
  - decompor o problema principal em subproblemas (modularização)
  - refinar os subproblemas até encontrar problemas mais simples,
  - refinar até se chegar a um nível de detalhe que permita implementar o algoritmo na linguagem de programação
- A modularização permite dividir o programa em módulos, com subprogramas claramente delimitados, que podem ser implementados, separadamente, por diversos programadores de uma equipa
- **Passo-a-passo**, significa que cada passo deve ser terminado antes de se iniciar o próximo passo
  - exemplo: é impossível "somar dois valores" antes de executar completamente o passo de introduzir/receber aqueles valores

Características fundamentais de um Algoritmo

- **Finitude**
  - um algoritmo deve sempre terminar após um número finito de passos
- **Definição**
  - cada passo de um algoritmo deve ser definido com precisão
  - as ações devem ser definidas rigorosamente e sem ambiguidades
- **Entradas**
  - um algoritmo deve ter 0 ou mais entradas (os dados que lhe são fornecidos no início)
- **Saídas**
  - um algoritmo deve ter 0 ou mais saídas (os dados obtidos e que tem uma relação específica com as entradas)
- **Eficiência**
  - todas as operações devem ser suficientemente básicas de modo que cada uma delas possa ser, em princípio, executada com precisão num tempo finito por um ser humano

Representação de um algoritmo

- Existem várias formas de representar um algoritmo (**notação algorítmica**)
  - descrição narrativa (usando uma linguagem natural)
  - pseudocódigo (ou pseudo-linguagem)
  - fluxograma (ou diagrama de fluxo)
- Não existe consenso entre os especialistas sobre qual é a melhor maneira de representar um algoritmo
- No entanto, atualmente a maneira mais comum de representar algoritmos é através de fluxogramas e pseudocódigo

Notações algoritmicas

Descrição narrativa

- É uma representação diretamente numa linguagem natural, usando expressões que evidencie como o problema pode ser resolvido computacionalmente
- Não há um padrão da forma de representar um algoritmo
- Exemplo

Enunciado: calcular a soma de dois números e mostrar o resultado

1. inserir/ler um número
2. inserir/ler outro número
3. calcular a soma entre os dois números inseridos/lidos
4. mostrar/escrever o resultado da soma

Pseudocódigo

- É a forma de representar algoritmos, podendo já indicar alguns requisitos que as linguagens de programação necessitam, tais como os tipos de variáveis e as instruções (comandos) similares aos usados em linguagens de programação
- O algoritmo é descrito numa forma muito próxima das linguagens de programação de computadores
- É também uma linguagem estruturada, pois deve seguir uma estrutura (ou padrão) que possui um formato semelhante ao das linguagens de programação, facilitando a codificação em qualquer linguagem de programação

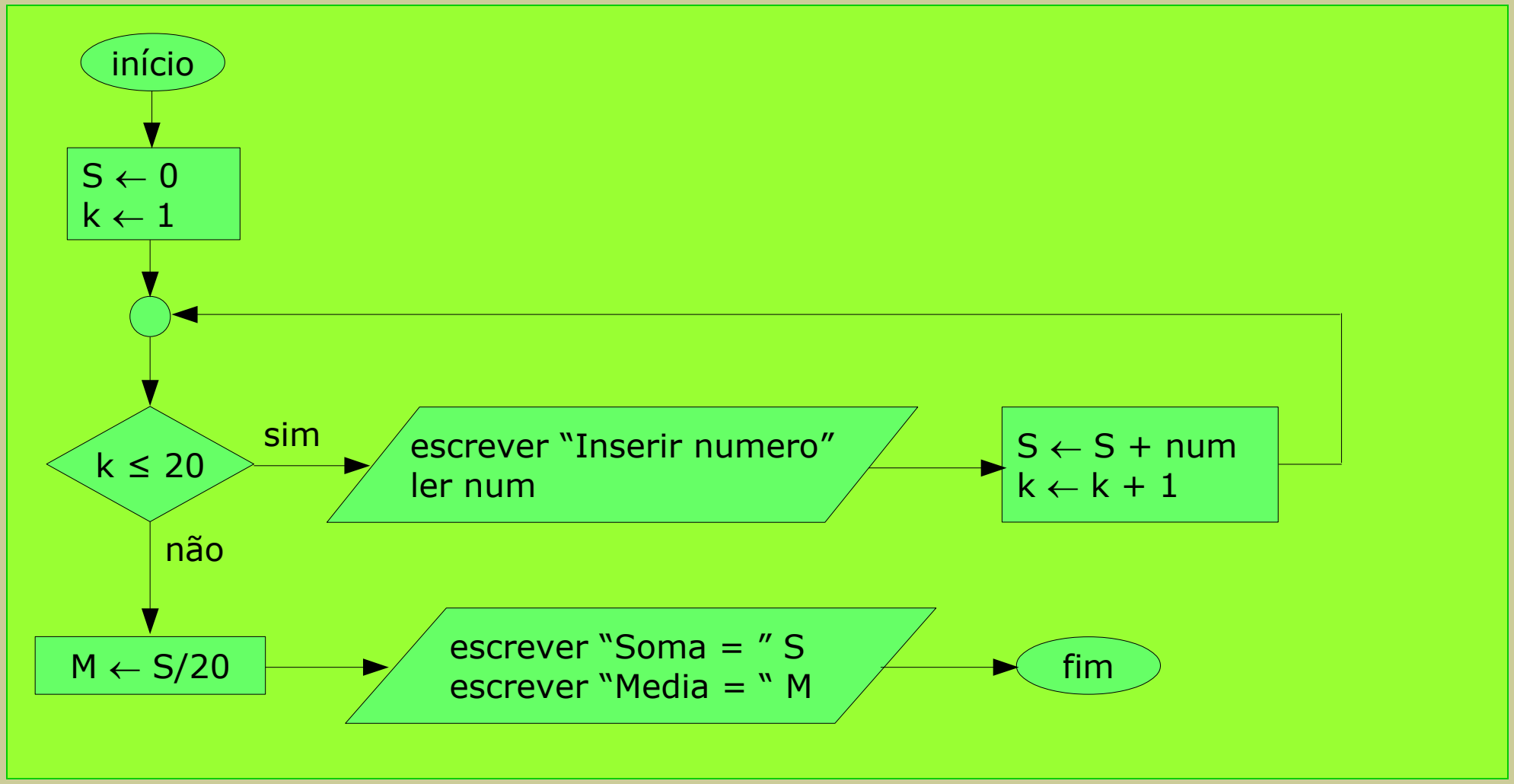
Pseudocódigo (exemplo)

Enunciado: determinar o maior elemento de um array de 1 dimensão com N números

```
algoritmo maiorNumeroArray1D (V, N)
  parâmetros de entrada: V (array 1D) e N (tamanho de V)
  parâmetros de saída: maior (maior número do array)
  maior ← V(1)       { o primeiro maior é o primeiro elemento do array V }
  k ← 2
  enquanto (k ≤ N) repetir
     se (V(k) > maior) então
        maior ← V(k)
     fim_se
     k ← k + 1
  fim_enquanto
fim_algoritmo
```

Fluxograma

- É uma representação gráfica de algoritmos, em que símbolos (ou formas) geométricos diferentes implicam ações (instruções, comandos) distintas
- Cada símbolo geométrico define a sua função genérica e no seu interior haverá uma descrição do passo do algoritmo, donde se inclui as representações das três estruturas básicas de controlo: sequência, seleção e repetição
- É uma forma intermédia entre a descrição narrativa e o pseudocódigo, mas é mais precisa do que a primeira

Fluxograma (exemplo)

Enunciado: inserir/ler 20 números e calcular a soma e a média entre eles