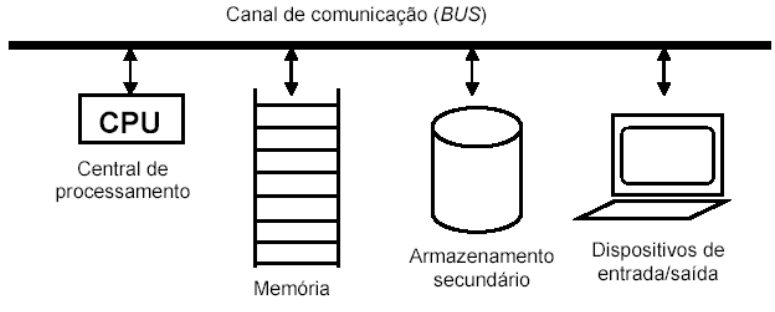
Computadores, Algoritmos e Linguagens

Modelo de um computador



CPU: Unidade Central de Processamento

Principal componente de um computador digital.

Localiza e executa as instruções de um programa.

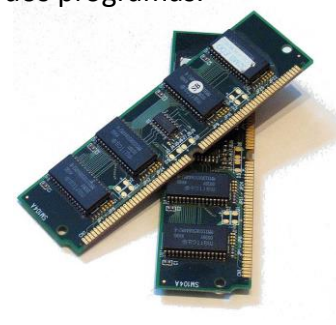
Capaz de executar operações simples com grande rapidez.

Memória principal (RAM)

Memória volátil usada para armazenar dados e programas.

Conteúdo modificável pelas instruções dos programas.

Permite acesso aleatório.



Disco rígido (HDD)

Um dos dispositivos não-voláteis mais usados para o armazenamento de grandes volumes de dados.

Utiliza superfícies magnetizáveis para armazená-los.

Permite acesso randômico aos dados armazenados.

Bits & Bytes

- Binários vs decimais
  - 0 = 0
  - 1 = 1
  - 10 = 2 ??
  - ?? = 8

1 Byte = 8 bits

1 kilobyte (KB ou Kbytes) = 1024 bytes

1 megabyte (MB ou Mbytes) = 1024 kilobytes

1 gigabyte (GB ou Gbytes) = 1024 megabytes

1 terabyte (TB ou Tbytes) = 1024 gigabytes

1 petabyte (PB ou Pbytes) = 1024 terabytes

…

Solução de problemas

Problema

Andando pelo campus alguém lhe pergunta:

- ✓ Como faço para ir ao centro da cidade?

O que você responde?

Problema

Faltam detalhes

- ✓ meio de transporte
  - ✓ envolve tempo e dinheiro
- ✓ endereço específico
- ✓ quando?
  - ✓ envolve trajeto devido a engarrafamento
- ✓ número de pessoas

Fica difícil responder sem saber as reais necessidades de quem perguntou

Solução de problemas

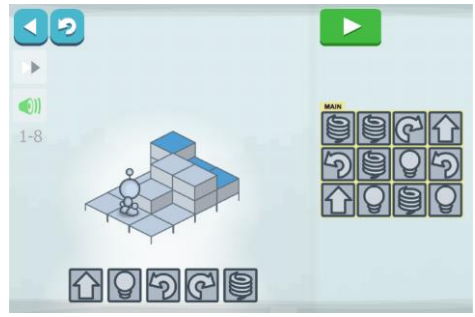
META: Encontrar uma solução algorítmica

Guia para construção de algoritmos

1. Compreender completamente o problema a ser resolvido, destacando os pontos mais importantes e os elementos que o compõem.
2. Definir os dados de entrada (ou simplesmente entrada), ou seja, quais dados serão fornecidos e quais elementos fazem parte desse cenário-problema.
3. Definir os dados de saída (ou simplesmente saída), ou seja, quais dados serão gerados depois do processamento.
4. Definir o processamento, ou seja, quais cálculos serão efetuados e as restrições a eles associadas. O processamento é responsável pela transformação dos dados de entrada em dados de saída.
5. Construir o algoritmo
6. Testar o algoritmo realizando simulações.

Algoritmo: conceito

Sequência finita e **não ambígua** de passos que permite a solução de um problema de maneira **automática** e **repetitiva**

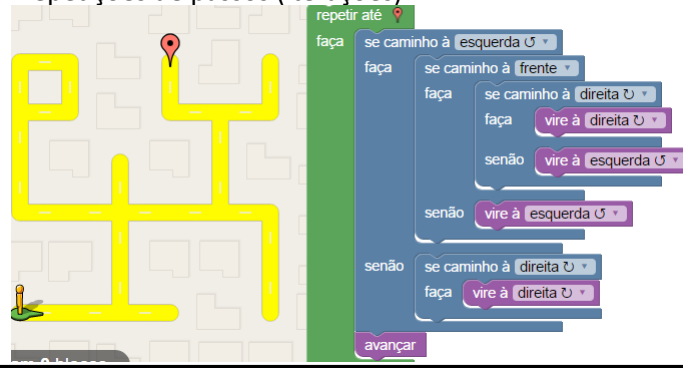


- **A linguagem utilizada na solução deve ser compreendida pelo executor da solução**

Algoritmo: componentes

A sequência de passos que **soluciona um problema** (algoritmo) é composta por:

- ✓ instruções diretas
- ✓ tomadas de decisões (em função do contexto atual)
- ✓ repetições de passos (iterações)



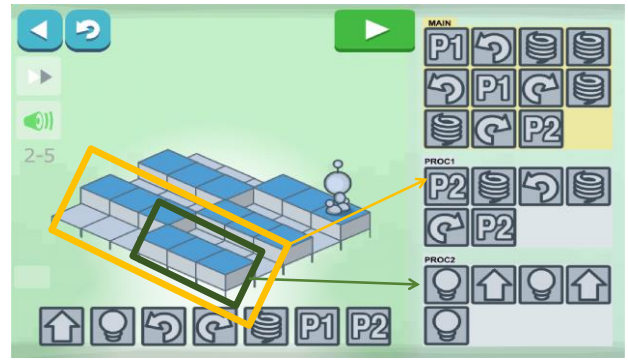
Algoritmo: procedimentos

Dentro do problema há uma série de problemas mais simples (tarefas).

Subsequências de instruções se repetem ao longo do solução !!!!

↓

Um modo de eliminar essa redundância é pela divisão em **módulos (procedimentos)**



Cada tarefa é programada em um procedimento. O módulo principal gerencia a sequência de execução das tarefas. Os demais módulos são ativados quando a execução de suas tarefas são requeridas.

Modularização

Decompor uma solução em uma série de subsoluções

Facilita a construção de uma solução ao dividi-la em partes menores → cada procedimento (módulo) é responsável pela resolução de uma parte do problema.

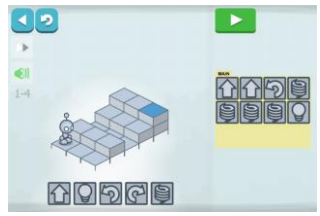
Cada módulo executa uma tarefa menor e bem definida → agrupa ações comuns a outras partes da solução, que podem ser acionadas quando necessário.

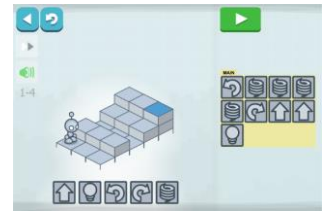
Algoritmo: exemplo

Sequência finita e não ambígua de passos, que permite a solução de um problema de maneira automática e repetitiva

Diferentes algoritmos podem solucionar um mesmo problema

- ✓ Exemplo : como se vestir de manhã?
  - ✓ Primeiro colocar a calça ou a camisa?
  - ✓ Meia e Sapato?





Algoritmos computacionais

Algoritmo: atores

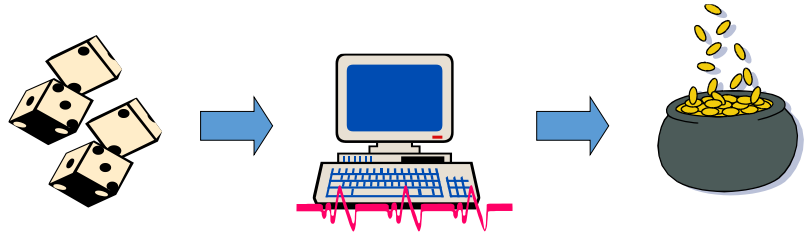
Atores envolvidos

- ✓ criador da solução: programador
- ✓ executor da solução:computador
- ✓ usuário da solução: humano

Linguagem utilizada na solução deve ser compreendida pelo executor da solução

Algoritmo computacional

Opera sobre um conjunto de valores de entrada de modo a gerar uma saída que seja útil para o usuário



Representação de algoritmos

Linguagem Natural

Pseudolinguagem

- ✓ Semelhante a uma linguagem de programação de alto nível, mas com regras gramaticais mais flexíveis.
- ✓ Apresenta as vantagens do emprego de um linguagem natural, mas com escopo mais restrito.

Linguagens de alto-nível

- Programar em linguagem de máquina é uma tarefa entediante e propensa a erros.
- A partir de meados dos anos 50 várias linguagens de alto nível foram criadas.
- Tais linguagens possuem nível de abstração relativamente elevados.
- Elas são mais próximas das linguagens utilizadas pelos seres humanos (linguagens naturais).

Linguagens de alto-nível

FORTRAN (1957)
COBOL (1960)
ALGOL (1968)
PASCAL (1970)
C (1972)
C++ (1983)
PYTHON (1991)
LUA (1993)
JAVA (1995)