

Lógica de Programação

Profª. Heloisa Moura

Arquitetura de um computador

Um computador é composto por duas partes principais:

- Hardware
- Software

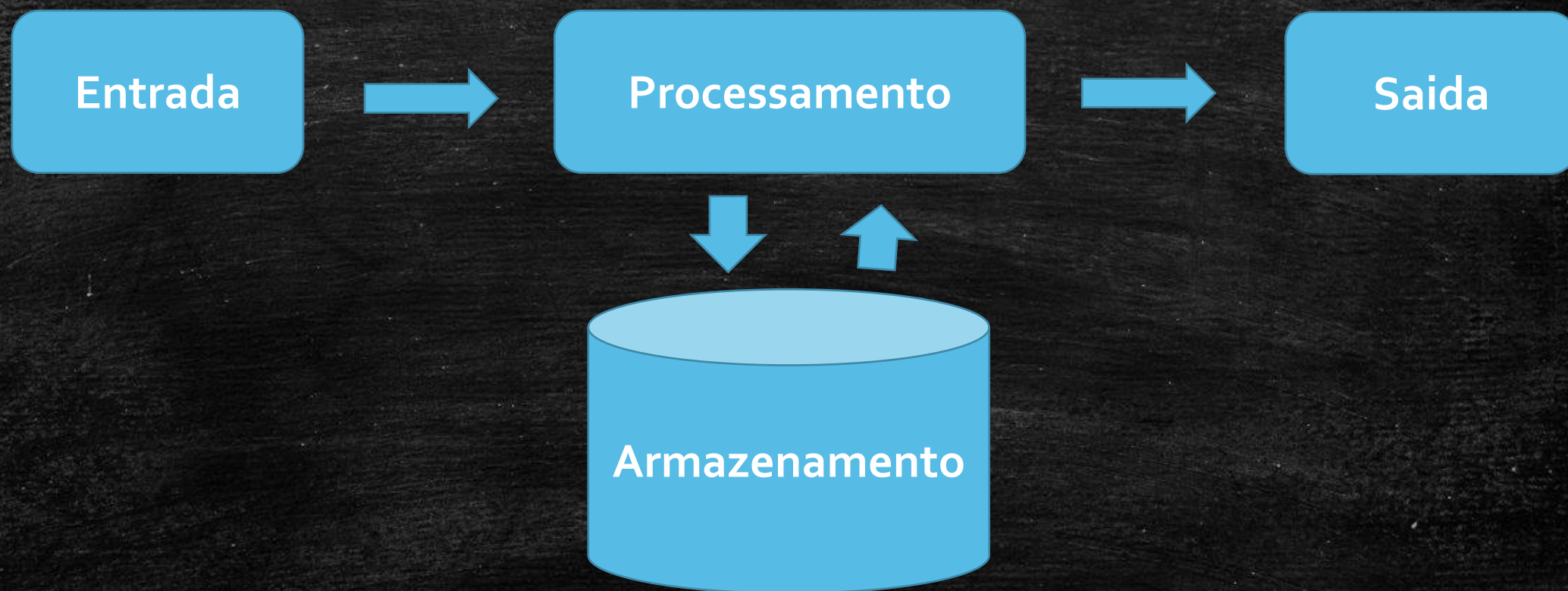
Hardware

Parte física de um computador composta pelos componentes e dispositivos; como monitor, teclado, CPU, memória RAM, discos etc.

Software

Parte lógica do sistema representada pelos programas que rodam sobre o hardware da máquina..

Processamento de Dados



Programa

Conjunto de instruções executada em uma ordem conveniente que contém as tarefas necessárias para resolver um problema usando o computador.

Linguagens de Programação

- Uma Linguagem de programação é um conjunto de regras utilizadas para criar programas de computador.
- Essas regras são usadas para expressar os algoritmos de forma precisa.
- Os algoritmos são escritos em uma linguagem de programação gerando um código fonte.
- Há três níveis principais de linguagens de programação:

Linguagens de Programação

- Linguagem de máquina: linguagem compreendida pelo computador. As instruções são representadas por valores binários (bits), 0 e 1.
- Linguagem de baixo nível: linguagem que utiliza instruções próximas a linguagem de máquina, porém compreensíveis por seres humanos. Ex.: Assembly.
- Linguagem de alto nível: linguagem cuja as instruções estão bem próximas da compreensão humana. Ex: C#, Python, Visual Basic, Java, Lisp e PHP.

Linguagens de Programação

- Uma linguagem de programação para ser efetivamente transformada em um programa e executada, deve ser convertida em código de máquina ("compilação") ou interpretada ("interpretação")

Compilação

- Método que traduz todo código-fonte em um ou mais arquivos que podem ser armazenados e executados quantas vezes se desejar, sem necessitar de nova compilação.
- Usa um compilador (SOFTWARE)
- Exemplo linguagem Java, c .

Interpretação

- Os comandos necessários contidos no código-fonte vão sendo executados à medida em que são traduzidos, sendo necessária a interpretação toda vez que se quiser rodar o programa.
- Usa um interpretador (Software)
- Exemplo: Java, Python

Lógica de Programação

O que é lógica de programação?

- Lógica é a ciência do raciocínio
- Encadeamento de pensamentos em uma sequência coerente – instruções
- Técnica para colocar instruções em uma sequência lógica para que seja possível atingir um determinado objetivo - em uma linguagem de programação.

Sequência Lógica

- Sequencia lógica: Passos ou tarefas executadas até atingir um objetivo. Solucionar um problema apresentado.
- Para que as tarefas possam ser realizadas com êxito, elas devem ser ordenadas.
- Tarefas ordenadas nos levam à noção de algoritmo.
- Mas antes, precisamos aprender a resolver problemas através da análise crítica.

Exemplo de sequencia lógica: comprar um refrigerante no mercado

- 1 Entrar no mercado;
- 2 Pegar um carrinho;
- 3 Ir até a seção de refrigerantes;
- 4 Pegar uma garrafa de refrigerante na gôndola;
- 5 Colocá-la no carrinho;
- 6 Dirigir-se até a fila do caixa;
- 7 Pagar pelo refrigerante;
- 8 sair do mercado.

Verificar se essa sequência: está certa?

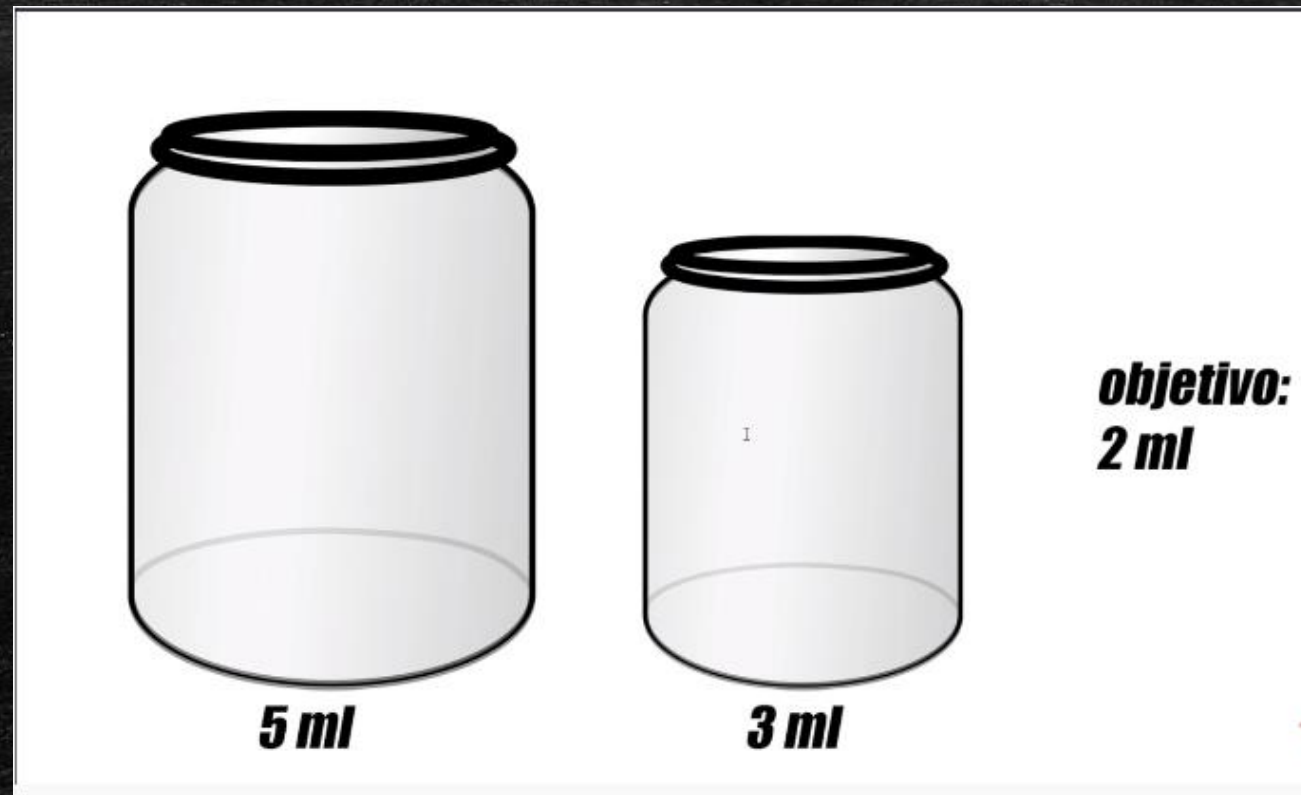
- 1. Pegar o carrinho
- 2. Dirigir-se a fila do caixa
- 3. Sair do mercado
- 4. Pegar uma garrafa de refrigerante na gôndola
- 5. Entrar no mercado
- 6. Colocá-la no carrinho
- 7. Pagar pelo refrigerante
- 8. Ir até seção de refrigerantes

Aprenda resolver problemas através da análise crítica

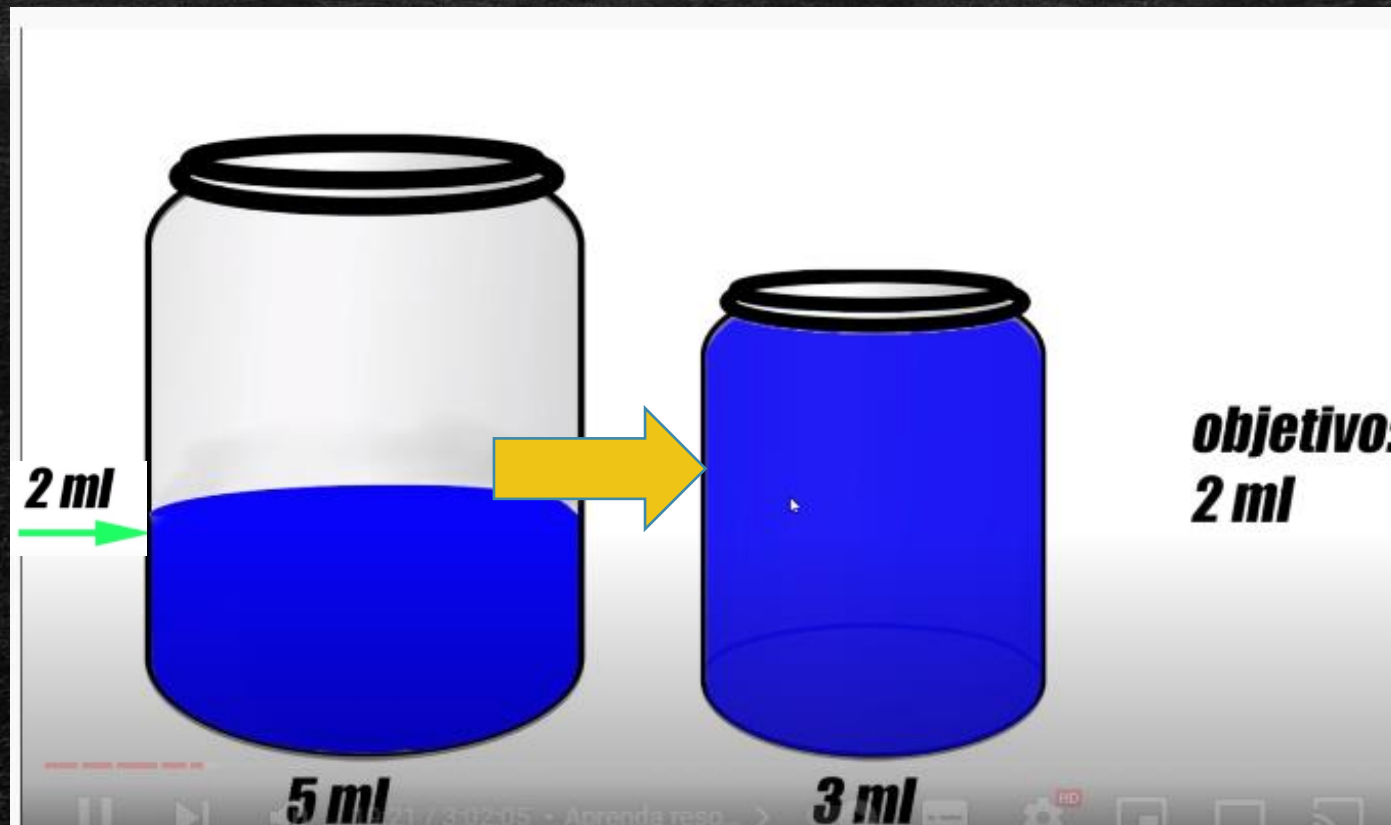
- “ Saber analisar e resolver um problema é mais importante que decorar os comandos de uma linguagem de programação!”

Aprenda resolver problemas através da análise crítica

Problema #1



Aprenda resolver problemas através da análise crítica



Aprenda resolver problemas através da análise crítica

Problema #2 Meu salário por hora

Escreva um programa que retorne o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas no mês.

????

Temos que fazer o seguinte cálculo:

Salário mensal / total de horas trabalhadas

Solução

salário mensal: R\$ 2500,00

horas trabalhadas: 176 horas

$2500/176 = \text{R\$ } 14,20$

salário hora = R\$ 14,20

“ Resolvi esse problema antes de escrever uma linha de código”

O mais importante é entender o problema, entender quais passos são necessários para resolver aquela situação.

Algoritmo

- Sequência de instruções ordenadas, finita e não ambígua, que são executadas mecanicamente para executar uma tarefa. Como uma “receita de bolo”.
- Um algoritmo deve ter fim, pode se comunicar com o mundo exterior e executar suas instruções em um tempo finito.
- Cada linha de um algoritmo deve executar uma única instrução.

Mundo exterior: Capturar informações do ambiente e devolver informações.

Ambígua: Não deixar dúvidas sobre o que ela faz, tem que ser direta.

Fases de um algoritmo

Entrada, processamento e saída

- Onde temos:

- Entrada: são os dados de entrada do **algoritmo**.
- Processamento: são os procedimentos utilizados para chegar ao resultado final.
- Saída: são os dados já processados.

O método 5Qs para montar um algoritmo

Analise criticamente o problema e descubra:

1. Quais são os dados de entrada necessários?
2. O Que devo fazer com esses dados?
3. Quais são as restrições desse problema?
4. Qual é o resultado esperado?
5. Qual é a sequência de passos a ser feitas para chegar ao resultado esperado?

Exemplo do método 5Qs para montar um algoritmo

Problema #2 Meu salário por hora

1. Quais são os dados de entrada necessários?
 - Salário mensal
 - Horas trabalhadas por mês
2. O Que devo fazer com esses dados?
 - Devo usar eles para calcular o valor hora que um funcionário recebe usando o cálculo $\text{salário mensal} / \text{horas trabalhadas}$.
3. Quais são as restrições desse problema?
 - O valor deve ser entregue somente em formato de salário por hora.

Exemplo do método 5Qs para montar um algoritmo

Problema #2 Meu salário por hora

4 - Qual é o resultado esperado?

- Valor hora que o funcionário recebe.

5 - Qual é a sequência de passos a ser feita para chegar ao resultado esperado?

Perguntar quanto a pessoa ganha por mês;
guardar essa informação;
perguntar quantas horas ela trabalha no mês;
guardar essa informação;
calcular o valor hora da pessoa;
exibir o valor hora daquela pessoa.

Representação de um algoritmo

Como que se pode representar um algoritmo então?

Falamos de algoritmo, sequência de instruções, fases do algoritmo. E como represento isso no papel ou no computador?

- Há varias formas de se representar um algoritmo:
 - Descritivo
 - Fluxograma
 - Diagrama de Chapin
 - Pseudocódigo (Português estruturado ou portugol)

Exemplo de Algoritmo Descritivo

- Montar um algoritmo para preparar um café utilizando papel e filtro(coador) e uma caneca.

Resolução

1. Pegue um coador
2. Pegue o papel de filtro
3. Coloque no coador
4. Coloque o conjunto sobre uma caneca
5. Pegue o pó de café no armário
6. Coloque um pouco de pó no coador
7. Ferver a água
8. Despeje a água no coador
9. Espere a água toda escorrer
10. Retire o coador
11. Adoce o café
12. Sirva o café

Refinamento de ações

- Note que o algoritmo para fazer o café é funcional, porém há muitas outras ações que devem ser realizadas, e outras tantas que podem ser desdobradas em ações menores. Essas ações são chamadas de **não-primitivas**.
- Já ações que não podem ser desdobradas são chamadas de **primitivas**.

Exemplo de refinamento

- Ação: Ferver a água:
- Refinamento:
 - Pegar a caneca
 - Levar a pia
 - Colocar sob a torneira
 - Abrir a torneira
 - Encher a caneca de água
 - Fechar a torneira
 - Levar a caneca ao fogo
 - Acender o fogo

Os algoritmos que vimos, de fazer café e ferver a água, pode ser representados por alguma das formas de representação apresentadas.

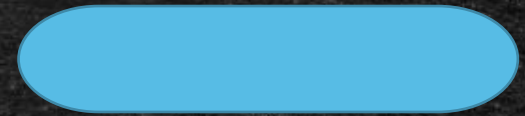
Algoritmo

Fluxograma

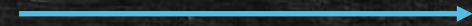
- Descreve o fluxo de dados e instruções usando desenhos geométricos básicos que simbolizam a entrada, saída e processamento de dados.
- Indicado para algoritmos de pouca extensão, ou se tiver um algoritmo com grande extensão, pode quebrá-lo em pedaços menores, pedaços modulares e desenhar fluxogramas para eles.
- Fluxograma é uma ferramenta interessante, principalmente quando se está começando aprender lógica de programação, porque ele permite visualizar o que está acontecendo no algoritmo na forma de figura.

Símbolos básicos para Fluxograma

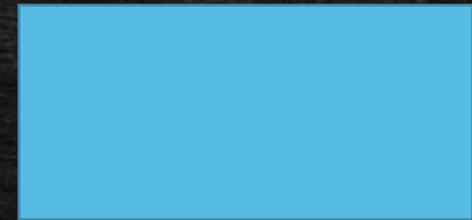
Início ou fim de programa ou fluxo



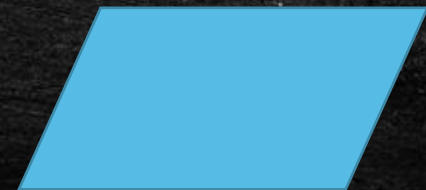
Seta de fluxo de dados



Bloco de processamento e manipulação de dados



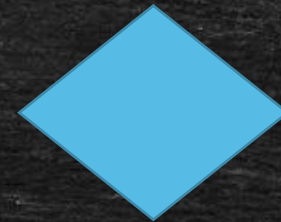
Entrada e saída de dados (via teclado)



Símbolos básicos para Fluxogramas

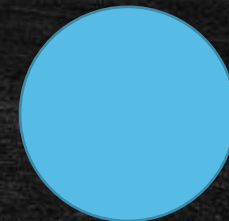
Estrutura de decisão

Quando no programa precisa decidir o que fazer naquele momento, e tem mais de uma alternativa possível.



Conector de partes do diagrama

Quando o diagrama é grande, pode cortar em partes, e a bolinha simboliza que ele terminou em um pedaço e continua em outro.



Símbolos básicos para Fluxograma

Conector de página

Quando o diagrama terminou e vai continuar na página seguinte.



Entrada manual de dados via teclado
Quando alguém vai digitar alguma informação que vai entrar para o programa.



Exemplo de fluxograma

Problema: Calcular a média de um aluno, baseado nas notas de quatro provas, e verificar se foi aprovado.

Critério: será aprovado se a média for igual ou superior a 7.0

Resolução 1: O algoritmo

Início

Entrar com as 4 notas

Calcular média

Verificar se média é maior ou igual a 7.0

Se sim, apresentar no vídeo "Aprovado"

Se não, apresentar no vídeo "Reprovado"

Fim

Resolução 2: fluxograma

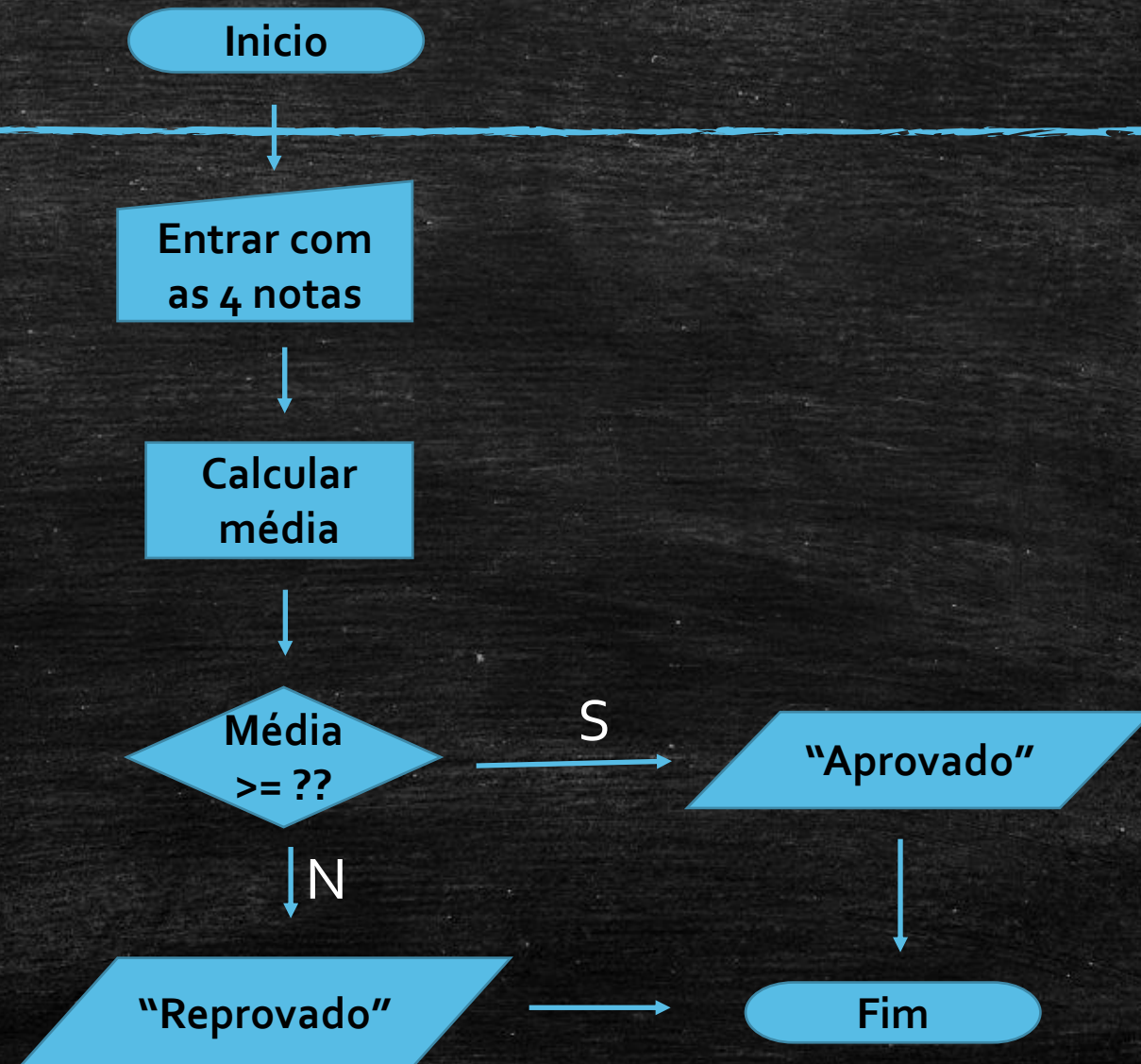
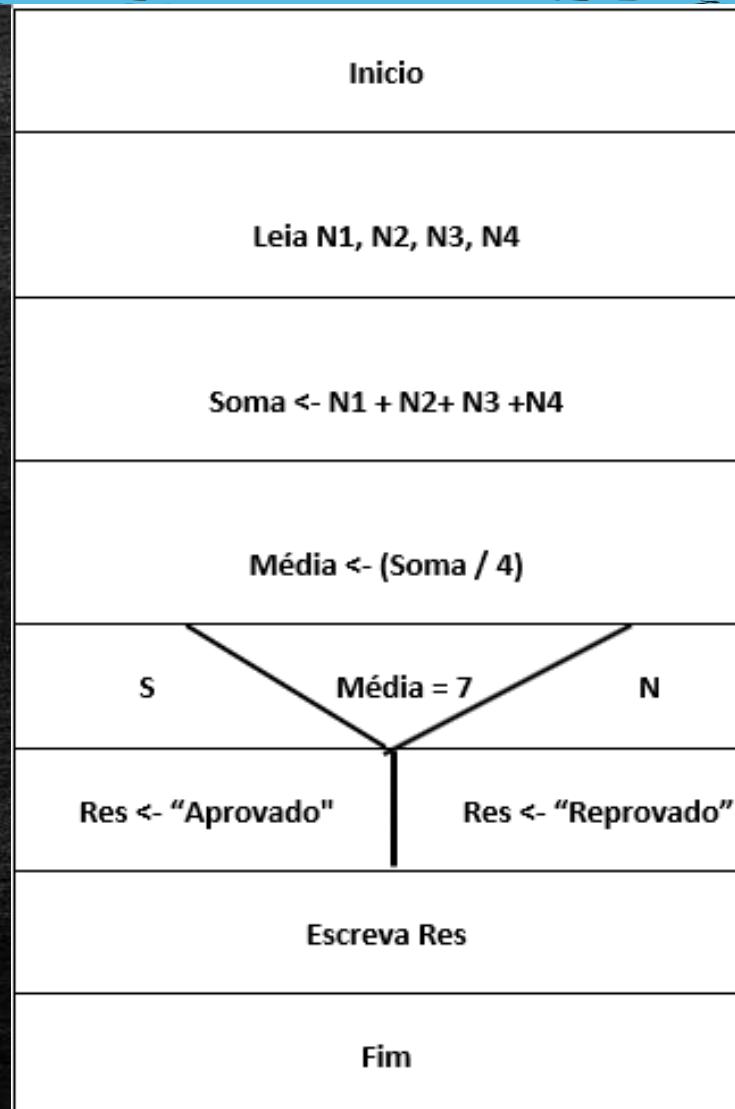


Diagrama de Chapin

- Desenvolvido por Nassim Shneiderman e ampliado por Ned Chapin.
- Substitui o fluxograma tradicional por um diagrama de quadros que apresenta uma visão hierárquica da lógica do programa.
- Facilita a codificação do algoritmo bem como a representação de certas técnicas, como a recursividade.
- Recursividade – Mecanismo básico para repetições.

Resolução 3: Diagrama de Chapin



Pseudocódigo (Português estruturado ou portugol)

- Técnica narrativa baseada em PDL – Program Design Language (Linguagem de Projeto de Programa).
- Notação para elaboração de algoritmos estruturados que serão utilizados para a criação e desenvolvimento em uma linguagem de programação qualquer.
- Também conhecido como Português estruturado ou Portugol.

Resolução 4: Pseudocódigo

algoritmo Média

Nome do
algoritmo

var

Declaração
de variaveis

res: caractere
N1, N2, N2, N3, N4: real
media: real

inicio

Tipo das
variaveis

leia (N1, N2, N3, N4)
media <- ((N1 + N2 + N3 + N4) / 4)

Bloco
principal
do
programa

se (media >= 7) então
 res <- "Aprovado"

senão

 res <- "Reprovado"

fimse

escreva ("Resultado= ", res)

fimalgoritmo

Pseudocódigo

algoritmo, var, inicio, fimalgoritmo : Palavras reservadas que fazem parte da linguagem de programação, e não pode ser usada fora do contexto.

Ex.: criar nome de objetos ou nome de variáveis com palavras reservadas.

E para testar??

Podemos testar nossos algoritmos, utilizando um software chamado VisualG. Ele é gratuito e permite testar diversos algoritmos em pseudocódigo. Também podemos testar com um compilador online gratuito, como por exemplo o Repl.it que nos permite testar pseudocódigos em linguagem de programação.

Exercícios

- Exercícios de Algoritmos (Em pseudocódigo)
COMANDOS DE ENTRADA E SAÍDA
- Questão_1: Escreva um algoritmo que exiba a idade de um usuário qualquer.
- Questão_2: Escreva um algoritmo que receba como entrada o total de minutos usados por um cliente e exiba o valor a ser pago por ele. Sabe-se que a lanhouse cobra R\$ 2,30 por cada hora de acesso(Lembrete: 1 hora = 60 minutos).
- Questão_3: Sabendo que a Copiadora da Faculdade Maurício de Nassau cobra R\$ 0,08 por cada cópia feita, escreva um programa que receba como entrada a quantidade de folhas de um livro e exiba o valor total a ser pago para copiá-lo. (Lembrete: cada folha corresponde a duas páginas, frente e verso).