



Sistemas Abertos vs. Fechados

Sistemas abertos: Interagem com o ambiente externo e podem trocar informações, energia ou materiais.

Sistemas fechados: Operam de forma independente, com pouca ou nenhuma influência do ambiente externo.

Uma empresa como sistema aberto

Recebe matéria-prima, transforma em produtos e vende no mercado. Ela interage com fornecedores e clientes.

Uma geladeira como sistema fechado

A geladeira mantém a temperatura interna controlada sem muita interferência externa.



Sistemas Abertos vs. Fechados

Um sistema operacional de código aberto (Linux) vs. fechado (Windows)

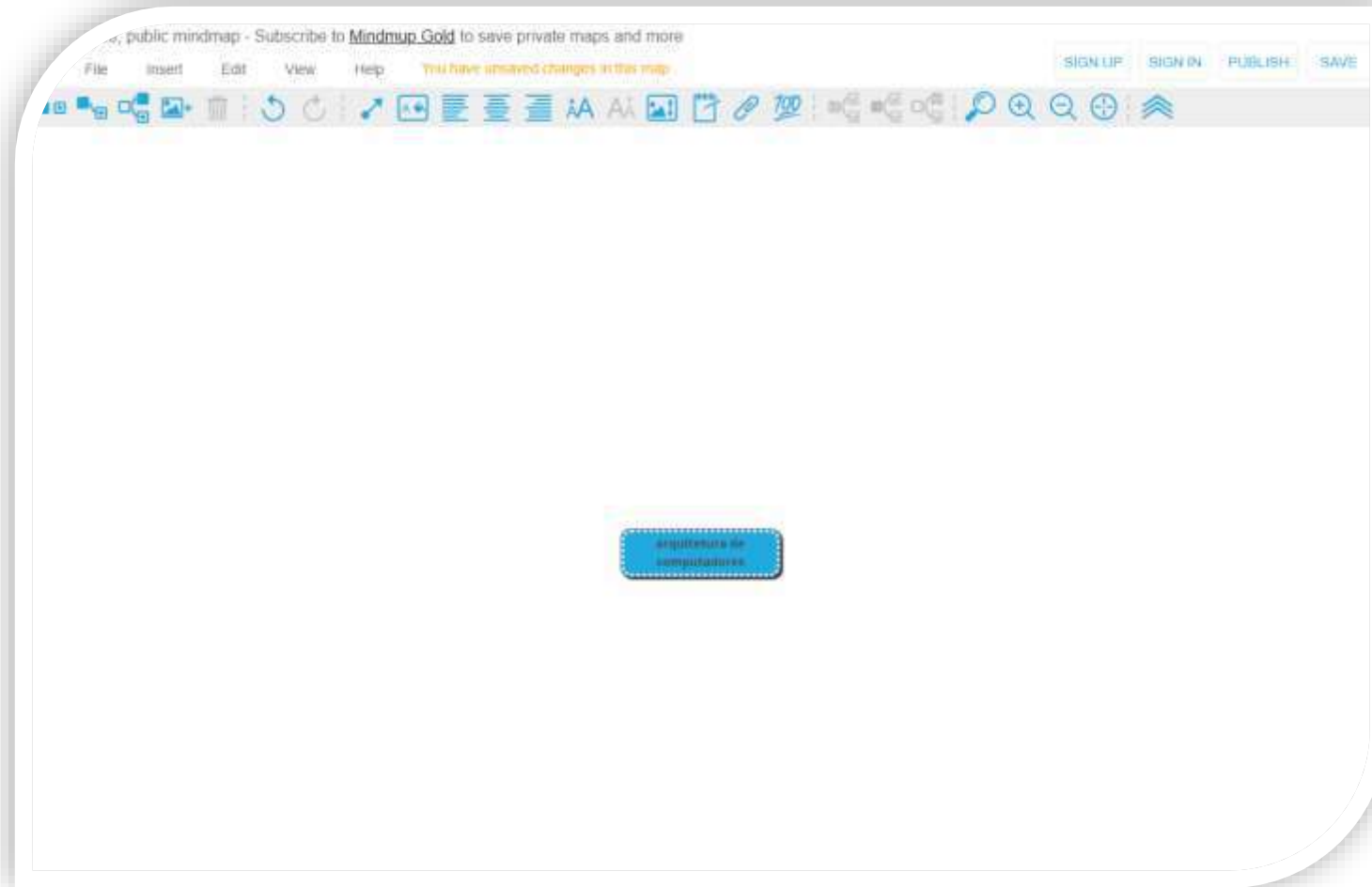


Windows é controlado pela Microsoft e tem código fechado.

Linux permite modificações e melhorias pela comunidade (aberto).



<https://app.mindmup.com/map/new/1743513134811>



Módulo 2 – Lógica de Programação

1

O que é

Lógica de Programação?

Sua importância no desenvolvimento de
software

2

Algoritmos

Definição e Características

3

Fluxogramas

Representação gráfica de
um algoritmo

4

Estruturas

Condicionais / Repetição

5

Prática

Escrevendo um algoritmo



O que é Lógica de Programação?

Algoritmo: É uma sequência finita de ações executáveis que visam obter uma solução para um determinado tipo de problema.

É simplesmente uma receita para executarmos uma tarefa ou resolver algum problema.

Conjunto de regras e técnicas (resolver problemas).

Habilidade de organizar comandos.

Ordem Correta; Coerência; Clareza; Eficiência.



- 1) Acordar;**
- 2) Levantar da cama;**
- 3) Tomar banho;**
- 4) Escovar os dentes;**
- 5) Tomar café da manhã.**

CURSO



Características de um Algoritmo

Finitude – Um algoritmo deve terminar em um número finito de passos.

Definição – Cada passo do algoritmo deve ser claro e preciso.

Entrada e Saída – Pode receber dados de entrada e gerar saídas.

Eficiência – Algoritmos devem ser eficientes para evitar desperdício de recursos.

Instruções / Condições



Por que Lógica de Programação é importante?

O computador não “pensa” por conta própria;

Lógica errada = o programa não funciona corretamente.

1) Levantar da cama;

2) Acordar; desordem

3) Tomar banho;

4) Escovar ambíguo

5) Se não perfeito,
repetir passo 4. ambíguo/repetição infinita

6) Colocar meia no pé esquerdo.

7) Colocar meia no pé direito. ineficiência



Exemplos de algoritmo

Receita de bolo

- Pegar os ingredientes (farinha, ovos, leite, fermento)
- Misturar todos os ingredientes.
 - Pré-aquecer o forno.
- Colocar a massa no forno e aguardar 40 minutos.
 - Retirar e servir.

Amarrar um cadarço

- Pegar os dois cadarços.
- Cruzar os cadarços e passar um por baixo.
- Formar um laço com um dos cadarços.
- Dar uma volta com o outro cadarço ao redor do laço.
- Puxar as pontas e ajustar.

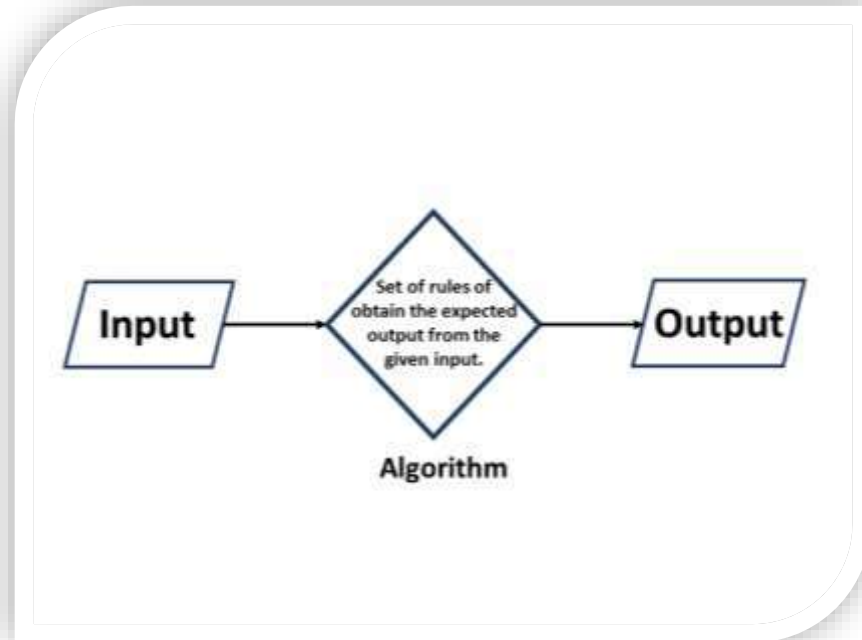
Sacar dinheiro no caixa eletrônico

- Inserir o cartão.
- Digitar a senha.
- Escolher a opção de saque.
 - Inserir o valor desejado.
 - Retirar o dinheiro.
- Pegar o recibo e finalizar.



Tipos de Algoritmo

Algoritmo Funcional (Interativo)



Depende de entradas do usuário ou de interações dinâmicas durante a execução.

Exemplo: calculadora □

Algoritmo Determinístico (Estático)



Dado um mesmo conjunto de entradas, sempre produzirá a mesma saída.

Exemplo: trocar lâmpada !



Aplicação de algoritmo

Ana (A), Beatriz (B) e Carol (C) foram almoçar juntas no restaurante japonês de robôs em Tokyo 🇯🇵

No final do almoço, como seria uma forma igualitária de todas pagarem a conta?

Vamos fazer um algoritmo bem explicado para o robô atendente.



Versão Algoritmo Determinístico (Estático)

Coletar o Valor Total da Conta.

O robô deve obter o valor total da conta do almoço. (TotalConta)

Divisão Igualitária.

O robô deve dividir o valor total pelo número de pessoas (3) = (ValorIndividual)

Exibir o Valor a Ser Pago por Cada Pessoa.

O robô informa para A, B e C que cada uma deve pagar o (ValorIndividual).



Determinístico: Para uma mesma entrada (valor total da conta), ele sempre produzirá a mesma saída (valor a ser pago por cada pessoa).

Estático: Ele segue um fluxo fixo de execução sem depender de interações contínuas com o usuário após a entrada inicial.



Versão Algoritmo Funcional (Interativo)

Determinar a quantidade de clientes.

O robô pergunta quantas pessoas estão na mesa.
(NºTotalClientes) Isso permite que o algoritmo funcione para qualquer número de clientes.

Coletar valores consumidos de cada pessoa.

Para cada cliente, o robô solicita o nome e o valor consumido, armazenando essas informações.
(consumo_{nome})

Cálculo do pagamento.

Calcula o total da conta, divide pelo (NºTotalClientes) e informa quanto cada um deve pagar.

Exibir o Valor a Ser Pago por Cada Pessoa.

Para cada cliente, o robô informa a quantia do que deve pagar (ValorIndividual).



Funcional (Interativo) porque permite entrada dinâmica de dados e pode ser reutilizado para diferentes mesas.

Repetição: Pode ser repetido várias e várias vezes. Sem problemas.



Exemplos de algoritmo (Atividade)

Em grupos, criem algoritmos que:

1. Escolher um restaurante para jantar
2. Definir um caminho para ir ao trabalho
3. Fazer um ovo frito
4. Preparar café
5. Organizar a agenda da semana
6. Escolher uma roupa para sair
7. Escolher uma sobremesa no self-service
8. Escovar os dentes
9. Planejar uma viagem de férias
10. Comprar um bilhete de cinema
11. Fazer compras no mercado
12. Ligar um carro



Obrigado!



    @fpftech.educacional