Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente1. Marque verdadeiro ou falso para as afirmações a seguir.

O raciocínio lógico só é usado na programação.

Verdadeiro

Falso

No algoritmo, o importante é completar as etapas, não importa a sequência.

A lógica é a primeira etapa da programação.

Algoritmos são usados no dia a dia.

Só existe um algoritmo correto para cada tarefa complexa.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

WEB APP

**3.** O sistema operacional é o software responsável por fazer a ponte, a interface entre o usuário e o hardware.

Associe corretamente as estruturas de funcionamento do computador de acordo com suas funções.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**4.** Um algoritmo descreve uma sequência de passos lógicos necessários para a execução de uma tarefa.

Considere as sequências narrativas, abaixo, de um algoritmo que descreve uma receita de bolo:

Sequência 1

1. Quebre os ovos.
2. Misture os ovos com o leite e a farinha.
3. Unte a forma.
4. Despeje a mistura na forma.
5. Ligue o forno.
6. Leve a forma ao forno.

Sequência 2

1. Ligue o forno.
2. Unte a forma.
3. Quebre os ovos.
4. Misture os ovos com o leite e a farinha.
5. Despeje a mistura na forma.
6. Leve a forma ao forno.

Identifique a alternativa correta:

Parte superior do formulário

1. Somente a sequência 1 está correta.
2. Somente a sequência 2 está correta.
3. A sequência 1 está parcialmente correta.
4. A sequência 2 está parcialmente correta.
5. As duas sequências estão corretas.

Parte inferior do formulário

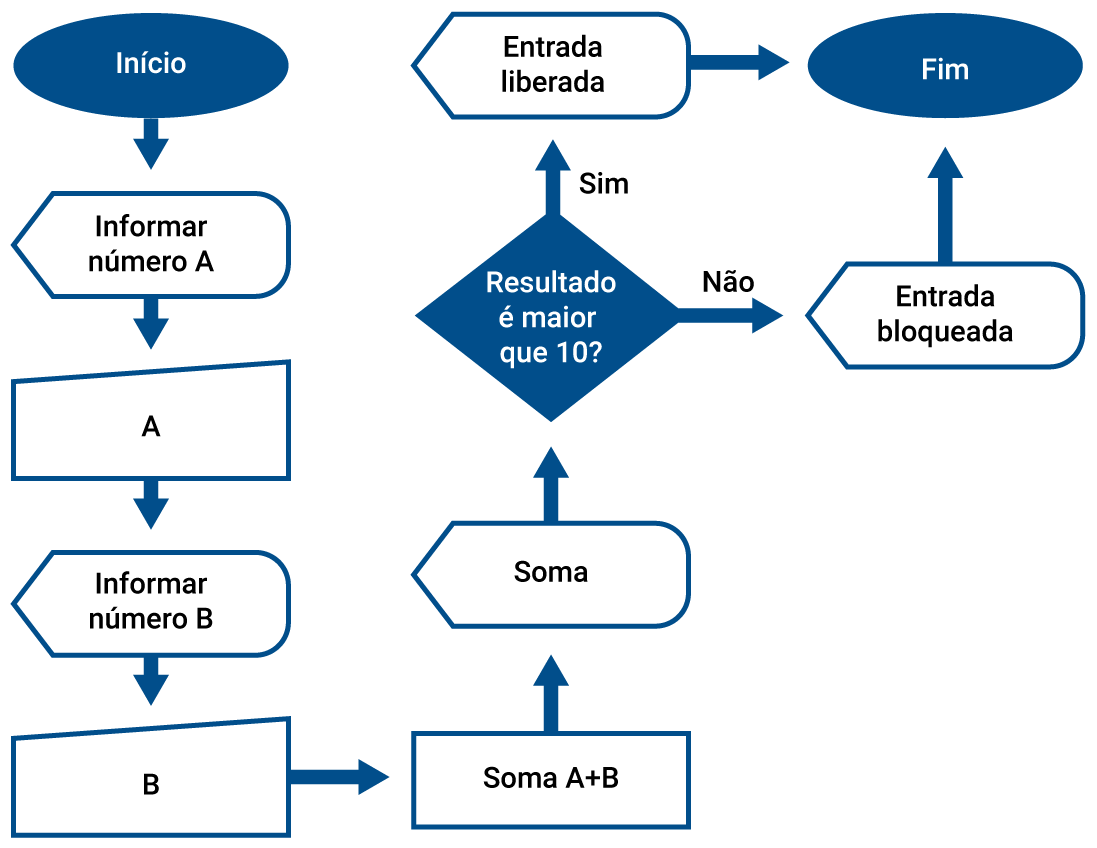
Uma imagem contendo Tabela

Descrição gerada automaticamente

**5.** O fluxograma a seguir descreve um algoritmo para verificação de uma senha. Aplique os valores abaixo, simule o passo a passo e identifique a alternativa que traz o resultado final correto.

Senha A = 18  
Senha B = 15

*Clique na imagem para ampliá-la.*

[[Desenho de um círculo

Descrição gerada automaticamente com confiança média](https://recursosdidaticos.senai.br/uploads/arquivo/64d53bfe91b399.99087343/img/aula-01/20017_MI_LogProg_u02_11-fluxograma-01_zoom.png)](https://recursosdidaticos.senai.br/uploads/arquivo/64d53bfe91b399.99087343/img/aula-01/20017_MI_LogProg_u02_11-fluxograma-01_zoom.png)

Parte superior do formulário

1. 34 - entrada liberada
2. 33 - entrada liberada
3. 03 - entrada bloqueada
4. 09 - entrada bloqueada

Parte inferior do formulário

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Gráfico

Descrição gerada automaticamente com confiança média

Gráfico

Descrição gerada automaticamente com confiança média

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Trabalha apenas com um arquivo, codificando o algoritmo linha após linha seguindo uma sequência lógica de evento. Exemplos: **C#**, **Delphi**, **Dart**, **Java**, **Kotlin**, **PHP**, **TypeScript**.

A programação estruturada é composta por três tipos básicos de estruturas, conforme você estudou:

* **sequências**: são os comandos a serem executados.
* **condições**: sequências que só devem ser executadas se uma condição for satisfeita (exemplos: if-else, switch, entre outros).
* **repetições**: sequências que devem ser executadas repetidamente até uma condição ser satisfeita (for, while, do-while etc.).

Trabalha com uma abordagem de reusabilidade de código, com o objetivo de simplificar a manutenção de um programa; é baseado na composição e interação entre diferentes unidades dentro do software. Exemplos: **C++**, **VB** **.NET**, **Java**, **Object Pascal**, **Objective-C**, **Python**, **SuperColider**, **Ruby**, **Dart**, **Smalltalk**, entre outras. As imagens a seguir mostram a diferença entre uma programação estruturada e uma orientada a objetos.

Uma imagem contendo Forma

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Imagem digital fictícia de personagem de desenho animado

Descrição gerada automaticamente com confiança baixa

**CONVERSÃO DE LINGUAGENS**

Como você já aprendeu, para executar qualquer tarefa, o computador necessita das instruções precisas e ordenadas, fornecidas pelos softwares. A ligação entre computador e software é feita por meio de uma linguagem de programação.

**Mas, como fazer essa "ponte", se a máquina usa uma linguagem de programação de baixo nível e o software usa uma linguagem de programação de alto nível?**

No início da computação, os códigos eram escritos de forma binária (0 e 1) em cartões perfurados, ou seja, cada comando gerava uma extensa sequência de 0 e 1. Na linguagem da máquina, um instrução corresponde a uma ação do computador. Imagine que cada marca ou modelo de computador tem sua própria linguagem, ou seja, a extensa sequência binária de um mesmo comando é diferente para cada marca ou modelo.

Era um processo complexo, extenso, lento e caro. Esse cenário levou à necessidade de otimizar e baratear toda a programação. Assim surgiram as linguagens de programação de alto nível e os processos de conversão de linguagem.

Imagem digital fictícia de personagem de desenho animado

Descrição gerada automaticamente com confiança baixa

Devido à diversidade de linguagens de programação de baixo e alto nível, há vários tipos de conversão.  
*Clique sobre as abas para conhecer as principais características de alguns tipos de conversão*.

Uma imagem contendo Aplicativo

Descrição gerada automaticamente