

Pensamento Computacional

2023/24

Apresentação UC



Apresentação



Objetivos de Aprendizagem

(conhecimentos, aptidões e competências a desenvolver pelos estudantes)

1. Reconhecer aspetos da computação no mundo à sua volta;
 2. Aplicar ferramentas e técnicas da computação para compreender e raciocinar sobre sistemas e processos tanto naturais como artificiais;
 3. Abstração, em diversas modalidades: modelação, decomposição, generalização e classificação;
 4. Compreender e explicar programas;
 5. Conceber e escrever programas;
 6. Encontrar e corrigir erros no código;
 7. Refletir aprofundadamente sobre o programa, incluindo avaliar a sua correção e adequação ao propósito;
 8. Compreender a eficiência do programa;
 9. Descrever o sistema a outras pessoas;
 10. Discutir aspetos como a inteligência e a consciência naturais e artificiais, a criatividade e a propriedade intelectual e as implicações morais e éticas da utilização de computadores.
-

Bibliografia

- Peter J. Denning, Matti Tedre. Computational Thinking, MIT Press, 2019.
 - John Guttag. Introduction to Computation and Programming Using Python: With Application to Understanding Data Second Edition. MIT Press, 2016. ISBN: 9780262529624
-

Metodologia de Ensino

- Baseadas em apresentação e discussão de problemas nas aulas teóricas e resolução de exercícios nas aulas práticas.
 - Serão utilizadas ferramentas desde as simulações em papel e lapis, à programação de robots infantis até à escrita e correção de programas com um IDE para Python.
-

Tipo de avaliação

- Avaliação distribuída **com** exame final.
 - Um teste (T1).
 - Nota mínima T1: 6 valores.
 - Nota mínima exame: 6 valores.
- Haverá um exame de recurso (EM) destinado aos alunos que desejem melhorar a sua classificação final e **para aqueles que não obtiveram os mínimos na avaliação normal.**

Classificação final

Nota Final = $(T1 + \text{Exame}) / 2$;

se $T1 \geq 6,0$ e $\text{Exame} \geq 6,0$

Obtenção de frequência

- São condições para a obtenção de frequência:
 - Cumprimento do limite de faltas legalmente estabelecido.
 - Obtenção de um mínimo de 10 (dez) valores na classificação final.

Pensamento Computacional

Aula 1



Pensamento Computacional



Pensamento Computacional



- Pensamento computacional ou pensamento algorítmico significa “**PENSAMENTO LÓGICO**”.
 - Tais denominações, remetem, obviamente, ao contexto computacional, pois derivam dos conceitos básicos da ciência da computação para solucionar problemas.
 - Mas, pensamento computacional ou pensamento algorítmico é apenas um **método organizacional para se chegar a uma resposta. É uma forma de pensar para a resolução de problemas.**
-

Quais as vantagens do “Pensamento Computacional”?

Planeamento

Capacidade
de pesquisa

Organização

Visão Crítica

Autonomia

Raciocínio
Lógico

Trabalho em
equipa

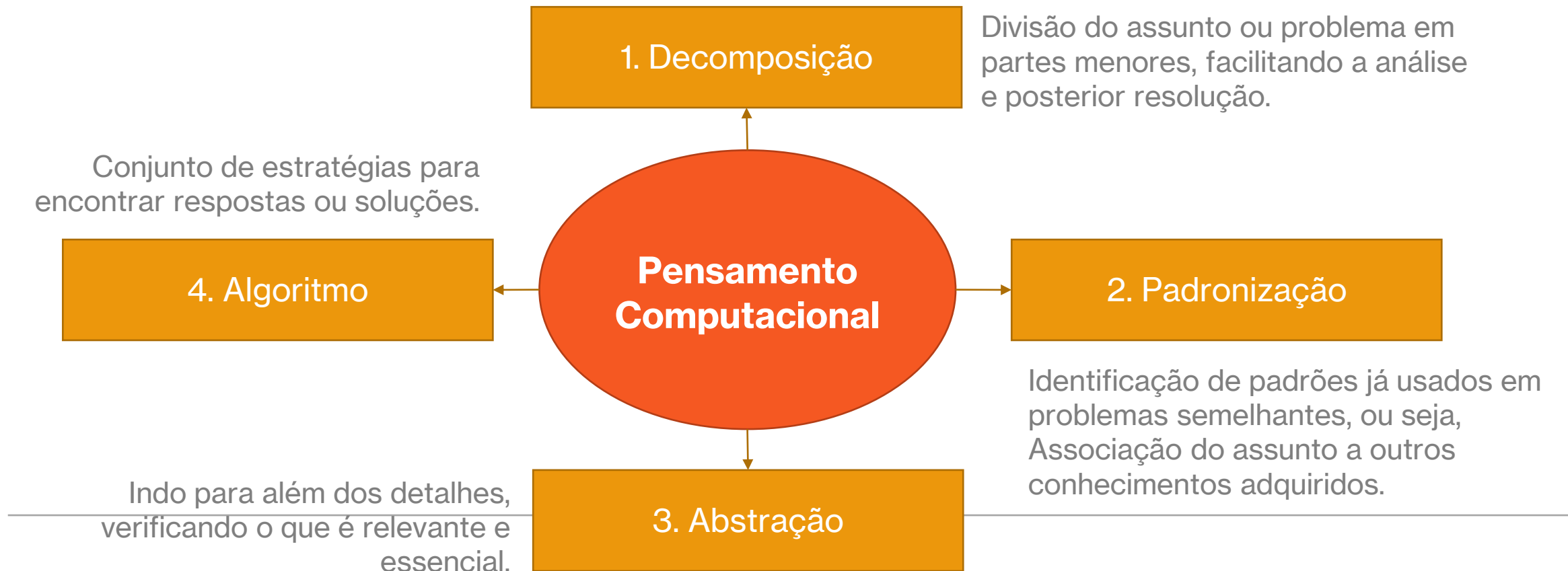
Participação

Criatividade

Inovação

As Fases do Pensamento Computacional

Perante uma situação/problema, pode agir-se seguindo 4 fases:



Pensamento Computacional na Prática

Lavar a Roupa na Máquina

1. Dividir as roupas por cores e tipos de tecido (**Decomposição**);
 2. Lembrar que, noutras vezes, já manchou roupas ou teve roupas com fiapos colados (**Padronização**);
 3. Manter o foco no essencial, ou seja, reconhecer os fatores que geram o problema: roupas escuras mancham roupas brancas e as escuras ficam com fiapos das brancas / toalhas soltam fiapos em todas as outras peças (**Abstração**);
 4. Estabelecer a ordem para execução da tarefa: em quantas vezes fará o processo de lavagem e como serão as fases para cada grupo separado e, assim, obter o resultado acertado de as roupas ficarem limpas e conservadas (**Algoritmo**).
-

Problema?

(latim *problema*, -atis, do grego *próblema*, -atos)

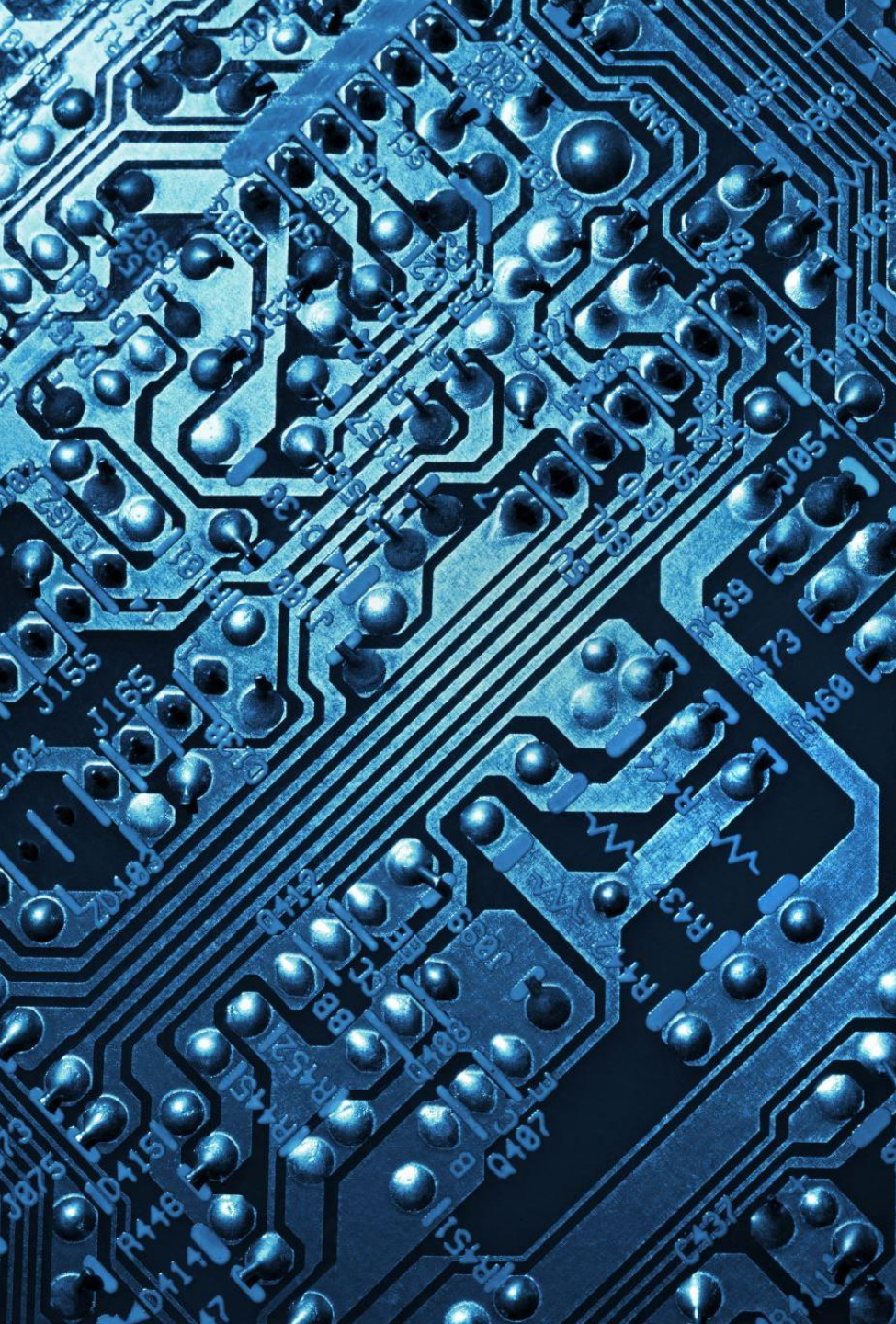
nome masculino

1. Questão matemática proposta para se lhe achar a solução.
2. Questão, dúvida.
3. O que é difícil de explicar.

"problema", in Dicionário Priberam da Língua Portuguesa [em linha], 2008-2021, <https://dicionario.priberam.org/problema> [consultado em 25-09-2022].

Lógica?

- A lógica é o ramo da Filosofia e da Matemática que estuda os métodos e princípios que permitem fazer distinção entre raciocínios válidos e não válidos, determinando o processo que leva ao conhecimento verdadeiro.
 - O uso da lógica é essencial na solução de problemas. Permite alcançar objetivos com eficiência e eficácia.
 - Ninguém ensina outra pessoa a pensar, mas a desenvolver e aperfeiçoar esta técnica, com persistência e consistência.
-



Algoritmo

- Ao utilizarmos a lógica para listar passos ordenados que resultam na solução de um determinado problema estamos a construir um algoritmo.
 - Ao contrário do que normalmente se imagina, o termo algoritmo não foi originado na computação e muito menos pode ser utilizado apenas no contexto computacional.
-



Algoritmo

- Podemos definir um algoritmo como:
 - uma sequência de passos que visa atingir um
 - objetivo bem definido;
 - uma sequência de passos bem definida que
 - deve ser seguida para a realização de uma tarefa;
 - ou solução de um problema.
-

Como ferver água?



Os passos **deste** processo de ferver água podem ser sumariados da seguinte forma:

- Encher uma panela com a quantidade desejada de água;
- Aumentar a fonte de calor para o máximo;
- Colocar a panela num dos discos do fogão;
- Colocar a tampa na panela;
- Manter a atenção na panela até se identificar grandes bolhas e serem formadas na água.

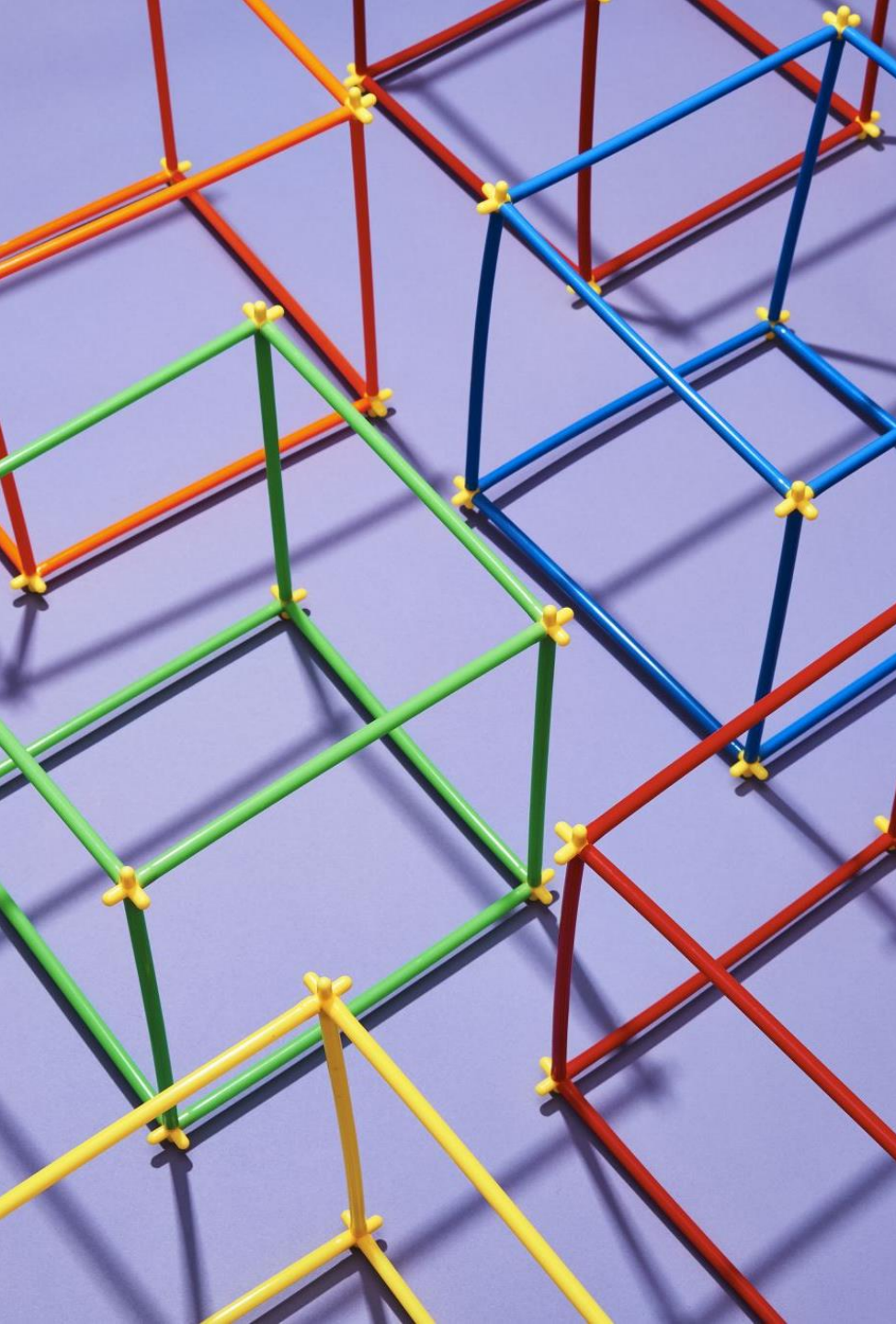
Este é um exemplo de um algoritmo.

Componentes de um algoritmo:

1. Início: especificar o ponto inicial do algoritmo;
 2. Entrada: definir os parâmetros de entrada do algoritmo;
 3. Tarefa (ou cálculo): Especificar os cálculos necessários;
 4. Saída: especificar os parâmetros de saída do algoritmo (resultados);
 5. Fim: especificar o momento final do algoritmo.
-

Exercício

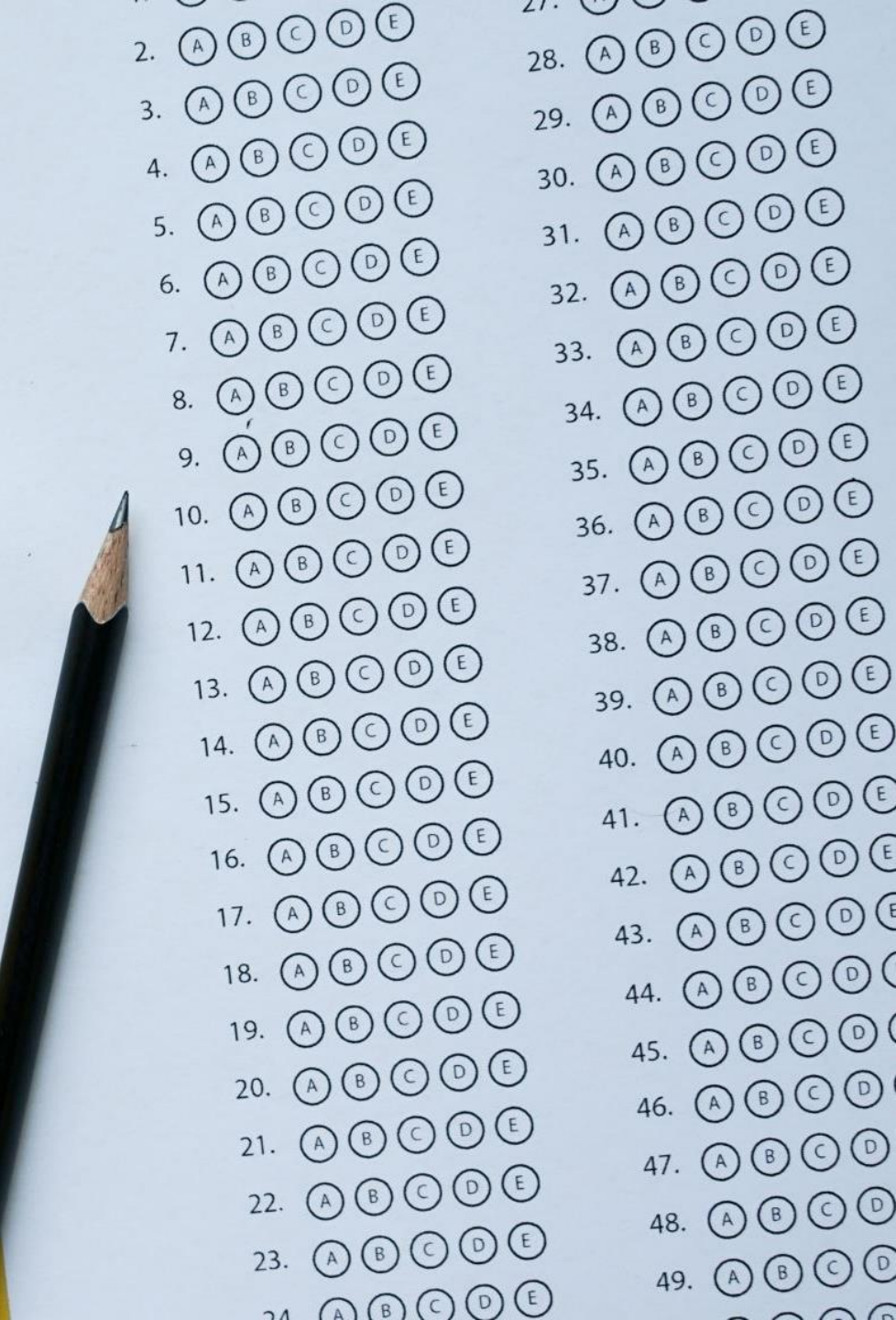
Definir um algoritmo para calcular e apresentar o salário bruto mensal de um trabalhador que trabalha à hora.



Formas de Representação de um Algoritmo

- Descrição Narrativa;
- Pseudocódigo;
- Fluxogramas.





Pseudocódigo

- Pseudocódigo é uma forma genérica de escrever um **algoritmo**, utilizando uma linguagem simples (nativa a quem o escreve, de forma a ser entendida por qualquer pessoa) **sem necessidade de conhecer a sintaxe de nenhuma linguagem de programação.**
-

Estrutura Básica

Algoritmo <nome do algoritmo>

Variáveis

<lista de variáveis>

Início

<bloco de comandos>

Fim

As palavras a negrito são palavras reservadas, palavras que indicam secções do pseudocódigo ou ações a serem executadas (instruções).

Problema do maior número

Linguagem natural?

Problema do maior número

```
Algoritmo Maior
Var
    num1, num2, maior: inteiro
Início
    Leia(num1, num2);
    se (num1>num2) então
        maior <- num1;
    senão
        maior <- num2;
    fimse;
    Escreva(maior);
Fim
```

Nome do algoritmo

Declaração de variáveis

Comando de entrada de dados

Comando de atribuição

Comando de saída de dados

Exercício

Calcular o salário de uma Pessoa após um aumento de 10%.

Exercício

Calcular o salário de uma Pessoa após um aumento de 10%.

- . Dados de entrada? Neste caso, o valor do salário.
 - . Saída? Neste caso, o novo salário.
 - . O processamento necessário para transformar a entrada na saída?
Neste caso, o cálculo do reajuste (10% do salário) e, em seguida, o acréscimo desse valor ao salário.
 - . $\text{Reajuste} = \text{salário} * 0.10;$
 - . $\text{Novo salário} = \text{salário} + \text{Reajuste}$
-

Exercício

Calcular o salário de uma Pessoa após um aumento de 10%.

Algoritmo NovoSalario

Variáveis

salario, reajuste, novosalario: Real

Início

Escreva "Digite o valor do salário";

Leia salario;

reajuste <- salario * 0.10;

novosalario <- salario + reajuste;

Escreva novosalario;

Fim
