

Aula 02

Introdução ao Pensamento Computacional



Algoritmos e Estrutura de Dados I

1º Semestre – CDN



Prof. Dr. Dilermando Piva Jr.

Conteúdo Programático - Planejamento

Conteúdo Programático		
Semana	Data	Temas/Atividades
1	12/08	Acolhimento e Boas vindas/ Introdução e Disciplina, Formas de Avaliação e Recursos Pedagógicos
2	19/08	Pensamento Computacional. O que é qual sua importância para Ciência de Dados
3	26/08	Primeiro Programa – Variáveis, Tipos de Dados e Saída em Python
4	30/08	/reposição/ Introdução a Computação e representação da informação – História e evolução da computação. A informação e sua representação. Conversão entre bases.
5	02/09	Operadores, Cálculo Simples e Entrada de Dados. Estruturas sequenciais.
6	09/09	Tomando Decisões: Estrutura Condicional (if/else)
7	16/09	Repetição de Ações: Estrutura de Repetição – Introdução aos loops (for e while)
8	23/09	Prática Integrada – Construindo um Jogo Simples (Adivinhação) – Pj1
9	30/09	Estruturas de Dados Parte 1 – listas e sequências (e strings)
10	07/10	Estruturas de Dados Parte 2 – Dicionários e Dados Estruturados
11	14/10	Integração – Projeto de Análise de Dados Simples com Listas e Dicionários – Pj2
12	21/10	Primeira Avaliação Formal. (P1). Correção da Avaliação após o intervalo.
13	28/10	Modularização do código – Introdução a Funções
14	04/11	Parâmetros, Retornos e Boas Práticas de Funções
15	11/11	Continuação de funções, Módulos e Pacotes em Python
16	18/11	Introdução a Machine Learning – Pj3
17	25/11	Semana de Apresentação PI de CDN
18	02/12	Introdução a Ciência de Dados – partes 1 e 2 – Pandas, Numpy, Pipeline e exemplos – Pj4
19	09/12	Segunda Avaliação Formal (P2). Correção da Avaliação após o intervalo
20	16/12	Exame / Avaliação Substitutiva. Divulgação do Resultado Final.

Exemplo Análise do Naufrágio do Titanic

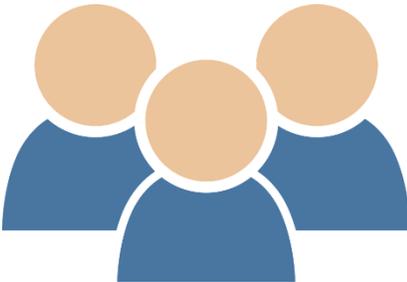
- Ocorreu em 15 de abril de 1912
- 1502 mortes dos 2224 passageiros e tripulantes



Nos
Pro...

Na última aula fizemos a implementação de uma solução, utilizando o computador e os programas (softwares)...

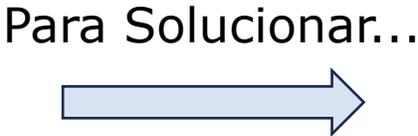
Implementação de uma Solução



Pessoas



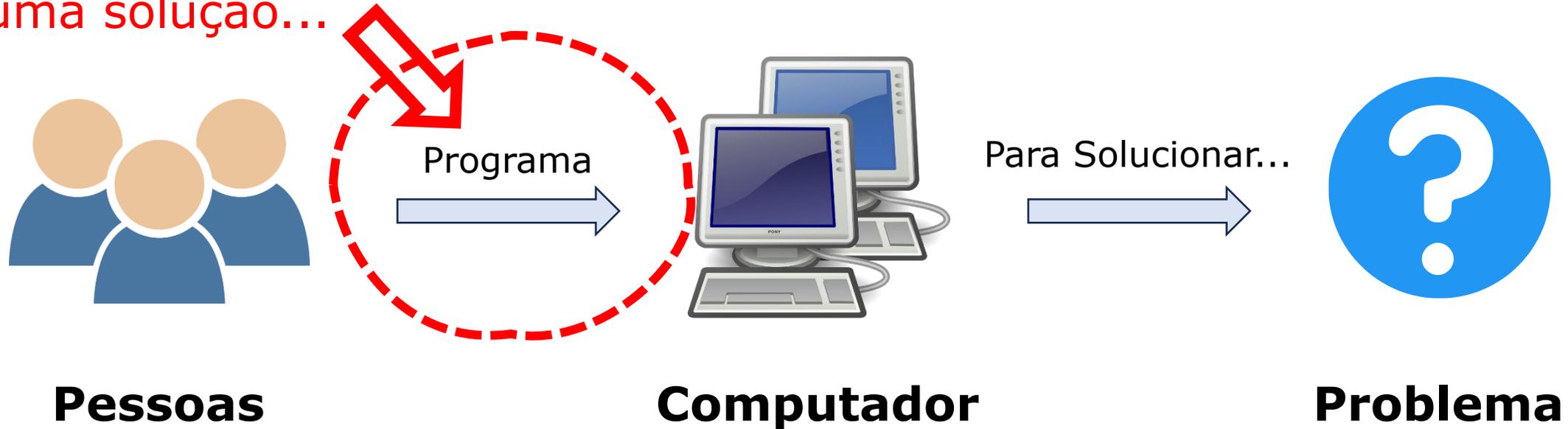
Computador



Problema

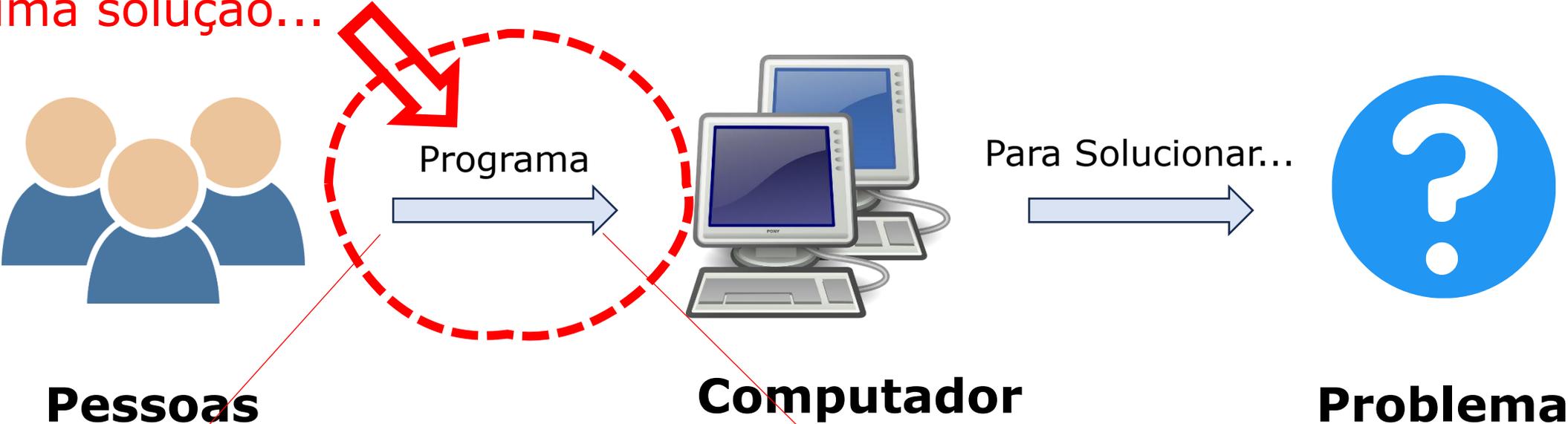
Implementação de uma Solução

A programação não é atualmente a primeira etapa para implementação de uma solução...



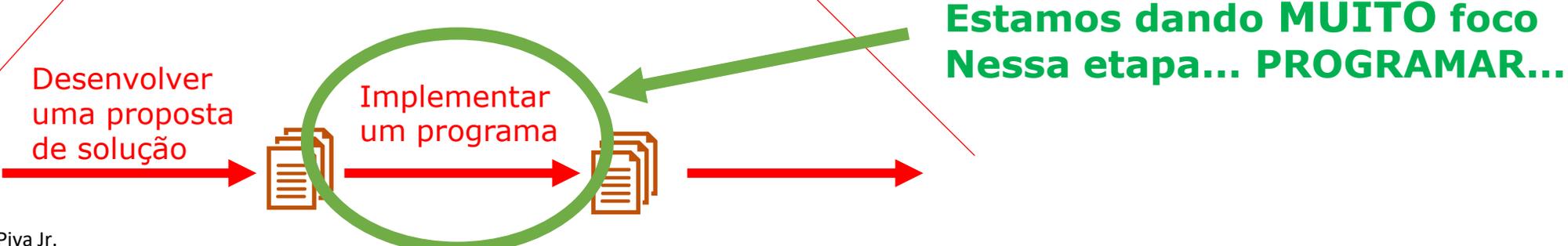
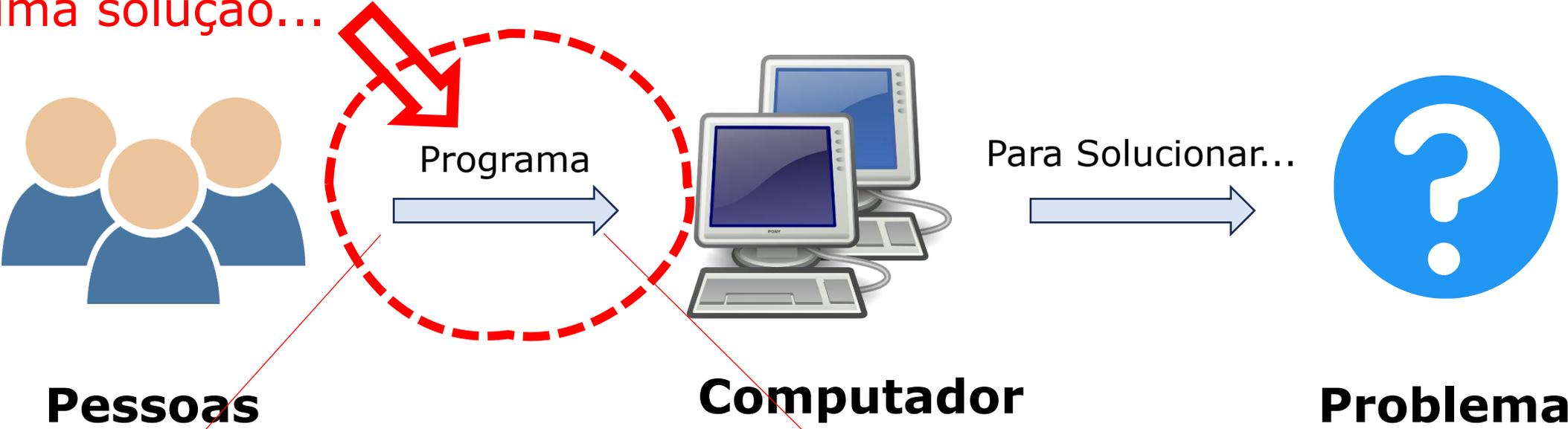
Implementação de uma Solução

A programação não é atualmente a primeira etapa para implementação de uma solução...



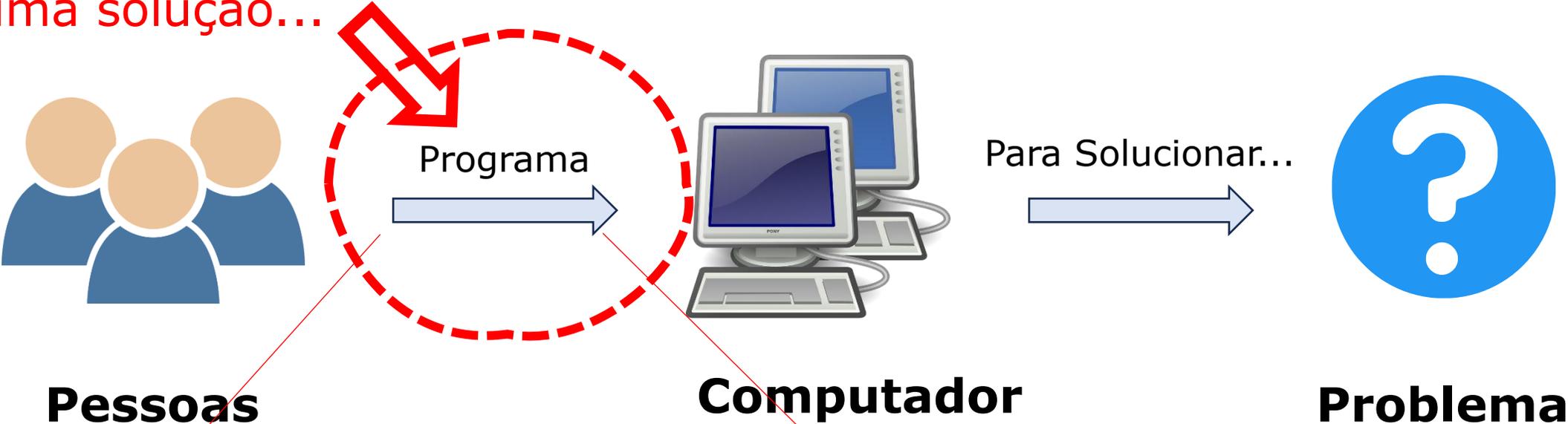
Implementação de uma Solução

A programação não é atualmente a primeira etapa para implementação de uma solução...



Implementação de uma Solução

A programação não é atualmente a primeira etapa para implementação de uma solução...

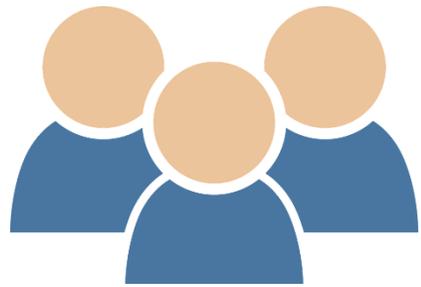


E muito POUCO foco nessa etapa ...

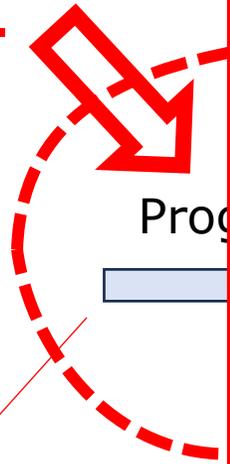


Implementação de uma Solução

A programação não é atualmente a primeira etapa para implementação de uma solução...



Pessoas



ASSIM...

Antes de escrever qualquer linha de código, já temos que ter em mente todo o processo do que vamos fazer (começo, meio e fim)... Quais dados vamos utilizar, qual abordagem implementaremos e quais os resultados esperados e como vamos apresentar tais resultados...

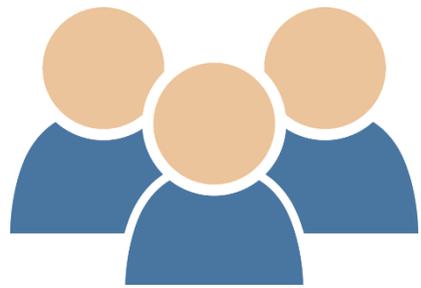
Desenvolver uma proposta de solução



Implementar um

Implementação de uma Solução

A programação não é atualmente a primeira etapa para implementação de uma solução...



Pessoas



PENSAMENTO COMPUTACIONAL...

Envolve tudo o que fazemos ANTES, DURANTE e DEPOIS da implementação... Da escrita de um programa de computador!

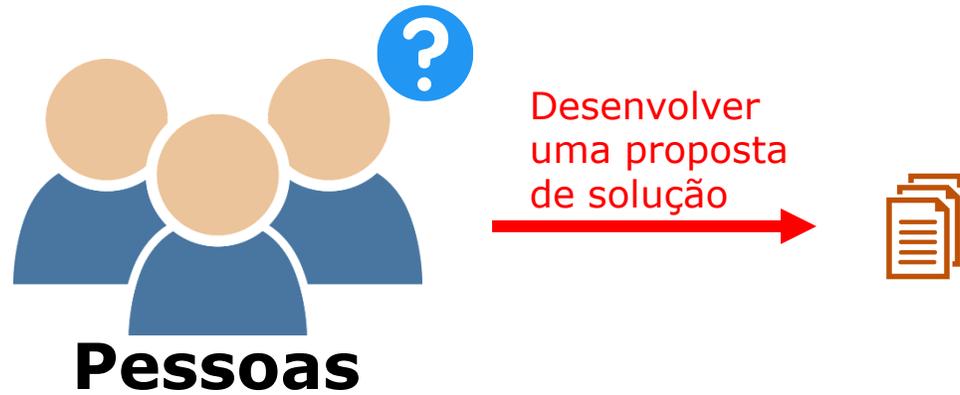
Desenvolver uma proposta de solução



Implementar um programa

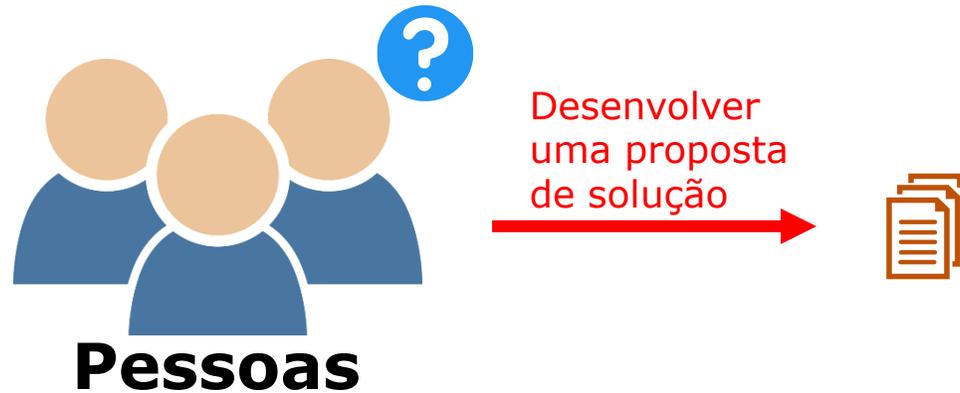


Implementação de uma Solução



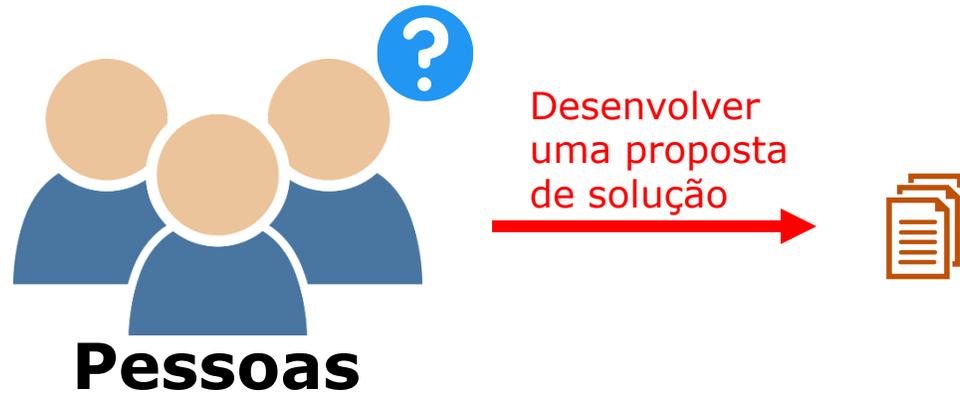
1. Devemos identificar: O problema tem “solução computacional”?

Implementação de uma Solução



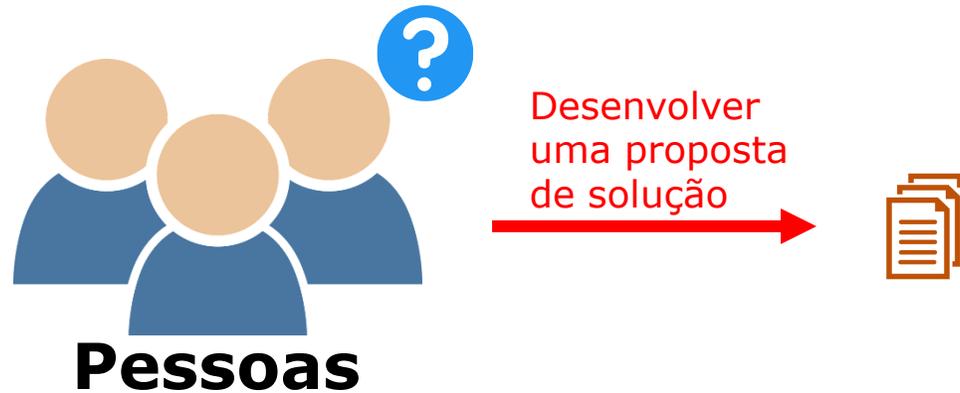
1. Devemos identificar: O problema tem “solução computacional”?
2. Podemos quebrar nosso problema em “sub-problemas” (em partes)?

Implementação de uma Solução



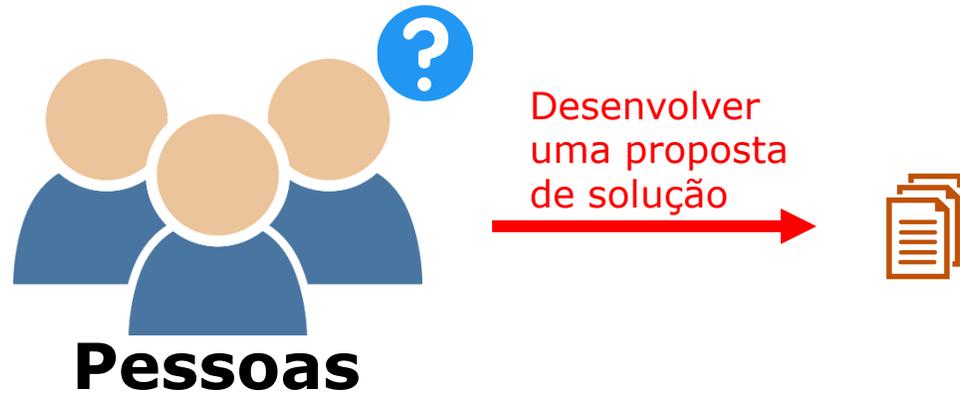
1. Devemos identificar: O problema tem “solução computacional”?
2. Podemos quebrar nosso problema em “sub-problemas” (em partes)?
3. Podemos perceber padrões ou características (nessas partes) que podem nos levar a potenciais soluções?

Implementação de uma Solução



1. Devemos identificar: O problema tem “solução computacional”?
2. Podemos quebrar nosso problema em “sub-problemas” (em partes)?
3. Podemos perceber padrões ou características (nessas partes) que podem nos levar a potenciais soluções?
4. Podemos identificar aspectos “não-essenciais” em nosso problema (ou em suas partes), que podemos ignorar?

Implementação de uma Solução



IDENTIFICAÇÃO DO PROBLEMA

1. Devemos identificar: O problema tem “solução computacional”?

DECOMPOSIÇÃO

2. Podemos quebrar nosso problema em “sub-problemas” (em partes)?

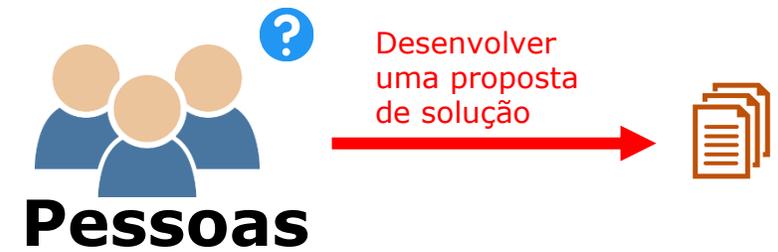
RECONHECIMENTO DE PADRÕES

3. Podemos perceber padrões ou características (nessas partes) que podem nos levar a potenciais soluções?

ABSTRAÇÃO

4. Podemos identificar aspectos “não-essenciais” em nosso problema (ou em suas partes), que podemos ignorar?

Implementação de uma Solução



IDENTIFICAÇÃO DO PROBLEMA

- Identificar se um problema é “solucionável” com um computador... Acredite ou não, por mais poderoso que um computador possa parecer, alguns problemas não podem ser necessariamente solucionado com computadores!

Implementação de uma Solução



Desenvolver
uma proposta
de solução



IDENTIFICAÇÃO DO PROBLEMA

- Identificar se um problema é “solucionável” com um computador... Acredite ou não, por mais poderoso que um computador possa parecer, alguns problemas não podem ser necessariamente solucionado com computadores!

230

A. M. TURING

[Nov. 12,

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO
THE ENTSCHEIDUNGSPROBLEM

By A. M. TURING.

[Received 28 May, 1936.—Read 12 November, 1936.]

The “computable” numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable *numbers*, it is almost equally easy to define and investigate computable functions

Implementação de uma Solução



Desenvolver
uma proposta
de solução



LIÇÃO DE CASA... NOTEBOOKLM

The screenshot displays the NotebookLM interface for a document titled "Números Computáveis: O Problema da Decisão". The interface is divided into three main sections: Fontes, Conversa, and Estúdio.

- Fontes:** A sidebar on the left containing a list of sources. The source "turing_1936.pdf" is highlighted with a red circle. Above the list are buttons for "+ Adicionar" and "Descobrir".
- Conversa:** The central area showing the document's content. The title "Números Computáveis: O Problema da Decisão" is prominently displayed. Below the title, it indicates "1 fonte". The main text discusses the document by A. M. Turing, titled "Sobre Números Computáveis, com Aplicação ao Entscheidungsproblem", and its focus on the theory of computability. It mentions Turing's exploration of the nature of computable numbers and the concept of "computational machines".
- Estúdio:** A sidebar on the right offering various tools and actions. It includes buttons for "Resumo em Áudio", "Resumo em Vídeo", "Mapa mental", and "Relatórios". Below these are several generated notes or summaries, such as "Turing e a Revolução da...", "Guia de Estudo: Números...", and "Números Computáveis e o...".

At the bottom of the interface, there is a text input field with the prompt "Comece a digitar..." and a "Salvar nota" button. A search bar at the bottom contains the text "Qual é a essência do conceito de 'número computável' e sua aplicação central?".

Exemplos de problemas NÃO Computáveis

O Problema da Parada (Halting Problem)

Descrição: Dado um programa de computador e uma entrada, será que o programa vai parar (finalizar) ou vai rodar para sempre?

Por que não é solucionável? Alan Turing provou em 1936 que não existe um algoritmo geral capaz de decidir isso para todos os programas possíveis.

✓ **...em Ciência de Dados:** Isso explica por que às vezes modelos de ML “não convergem” e porque precisamos definir critérios de parada.

O Problema de Determinar a Verdade de Todas as Proposições Matemáticas (Problema de Entscheidungsproblem - Hilbert)

Descrição: Existe um algoritmo que pode dizer, para qualquer proposição matemática, se ela é verdadeira ou falsa?

Por que não é solucionável? Kurt Gödel (com seus teoremas da incompletude) mostrou que há proposições que são *indecidíveis* dentro de certos sistemas formais.

✓ **... em Ciência de Dados:** Nem todos os problemas podem ser totalmente “modelados” em um sistema lógico ou computacional.

Compressão Ótima de Dados

Descrição: Existe um algoritmo que pode, para qualquer arquivo, determinar a menor representação possível dele?

Por que não é solucionável? Isso está relacionado ao conceito de *complexidade de Kolmogorov*, que é indecidível.

✓ **... em Ciência de Dados:** Algoritmos de compressão como ZIP são heurísticos; eles não garantem a melhor compressão possível.

Decidir Se um Código-fonte É Livre de Bugs

Descrição: Existe um algoritmo que possa analisar qualquer código e dizer se ele tem ou não bugs?

Por que não é solucionável? Essa é uma generalização do problema da parada e também não tem solução geral.

✓ **... em Ciência de Dados:** Em pipelines complexos, a verificação completa de ausência de erros é impossível – usamos testes e validações para mitigar isso.

Implementação de uma Solução



Desenvolver
uma proposta
de solução



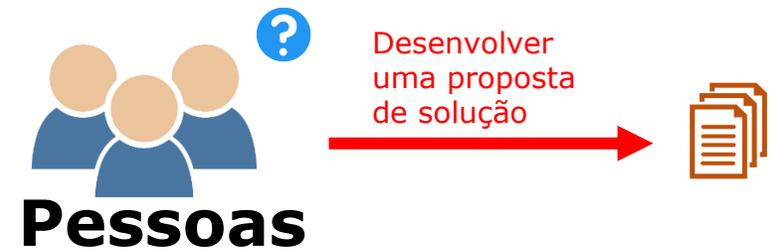
IDENTIFICAÇÃO DO PROBLEMA

- Identificar se um problema é “solucionável” com um computador... Acredite ou não, por mais poderoso que um computador possa parecer, alguns problemas não podem ser necessariamente solucionado com computadores!
- Como podemos saber quando um problema é um bom candidato para usar um computador para gerar um solução?
- Como nós podemos incorporar diferentes tipos de dados para solucionar esse tipo de problema?

Geralmente Problemas que envolvam muita subjetividade não são bons candidatos...

Exs: *Me diga qual é o filme mais engraçado que posso assistir em uma sexta-feira à noite?*
ou *Qual é o quadro mais bonito que posso pendurar na parede de meu novo quarto?*

Implementação de uma Solução

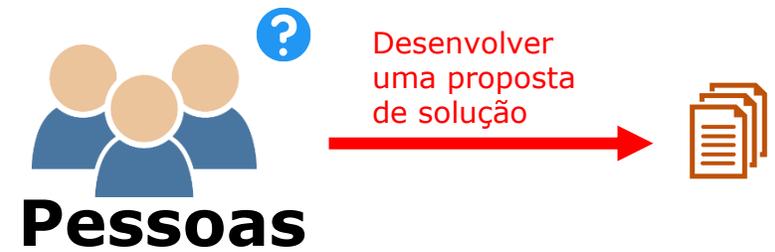


IDENTIFICAÇÃO DO PROBLEMA

Coisas que devemos pensar...

- **TÓPICO:** Qual é o grande problema que você está querendo resolver? É possível transcrever isso em algum tipo de “declaração do problema”.. Descrição detalhada? Em seguida devemos decidir se essa declaração é muito ampla ou muito vaga?
- **DADOS:** Que informação ou dados que poderiam nos ajudar ou contribuir para uma abordagem mais eficaz para solucionarmos esse problema? Que outras informações ou dados precisamos coletar para nos ajudar?
- **VIABILIDADE:** Podemos resolver o problema com o que sabemos? É solucionável com um computador e temos todos os recursos e informações necessários para resolver o problema?

Implementação de uma Solução



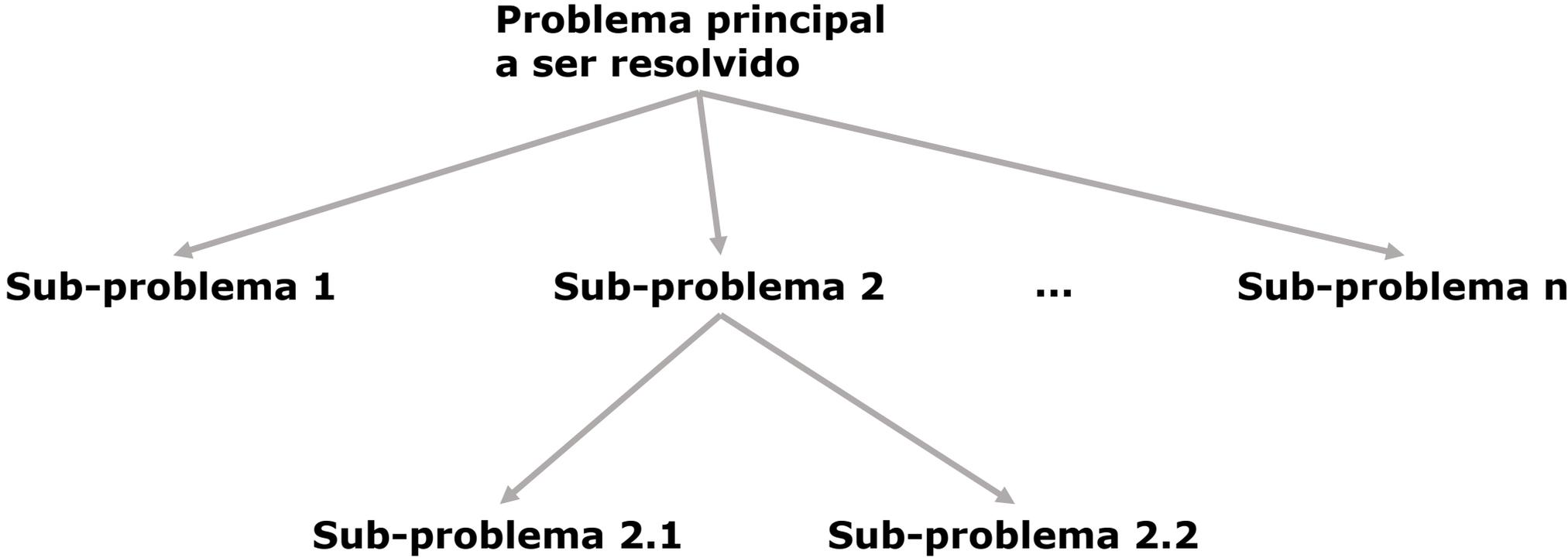
DECOMPOSIÇÃO

Depois de identificarmos um problema que achamos que podemos resolver usando um computador...

- Podemos quebrar nosso grande problema em partes menores... Em sub-problemas que são mais fáceis de resolver?
- Se a declaração de problema for muito grande, é difícil ver como você pode começar a desenvolver uma solução conceitual.
- Se você puder decompor seu grande problema em subproblemas cada vez menores, poderá começar a pensar em como resolver esses subproblemas específicos e menores.
- Resolver os subproblemas pode então ajudá-lo a chegar a uma solução para sua declaração de problema original (seu "problemão")
- Às vezes, você pode usar fluxogramas, uma estrutura em árvore ou outro organizador gráfico para ajudar com isso.

Implementação de uma Solução

DECOMPOSIÇÃO



Implementação de uma Solução



Desenvolver
uma proposta
de solução



RECONHECIMENTO DE PADRÕES

Podemos identificar padrões ou características familiares que possam nos levar a soluções potenciais?

Encontrar semelhanças ou padrões entre os problemas menores e decompostos pode nos ajudar a resolver esses problemas de forma mais eficiente.

Refleta sobre...

- Você já viu soluções para problemas semelhantes anteriormente que possamos usar aqui? Consegue pensar em problemas parecidos que já foram resolvidos antes e que poderiam ajudar a abordar estes problemas?
- De que forma este problema é igual ou diferente de outros problemas que você já identificou ou solucionou? E pode usar essas semelhanças para ajudar a chegar a uma solução?

Implementação de uma Solução



Desenvolver
uma proposta
de solução



RECONHECIMENTO DE PADRÕES

Análise de Sentimentos (Ciência de Dados)

Cenário: Você tem milhares de avaliações de clientes.

Padrão identificado: Palavras como "*ótimo, excelente, maravilhoso*" aparecem em textos com nota 5, enquanto "*horrível, péssimo*" aparecem com nota 1.

Aplicação: Usar esses padrões para treinar um modelo de classificação de sentimentos.

Filtragem de Spam em E-mails

Cenário: Um sistema precisa separar mensagens legítimas de spam.

Padrão identificado: Mensagens com palavras como "*ganhe dinheiro rápido*", muitos links e erros ortográficos costumam ser spam.

Aplicação: Criar filtros automáticos usando esses padrões.

Diagnóstico Médico Assistido por IA

Cenário: Classificar imagens de raio-X como "normal" ou "com pneumonia".

Padrão identificado: Certas áreas *opacas* nas imagens aparecem com frequência em pacientes com pneumonia.

Aplicação: Treinar um algoritmo para reconhecer esse padrão.

Implementação de uma Solução



Desenvolver
uma proposta
de solução



ABSTRAÇÃO

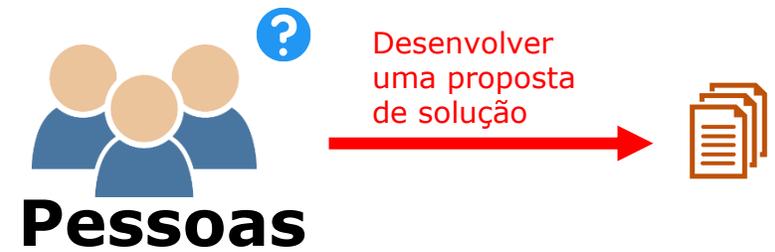
Podemos identificar aspectos não essenciais do nosso problema que podemos ignorar? Ou seja... Que estão nos distraindo.. E servindo como um desvio da solução do nosso problema principal?

Isso envolve filtrar ou ignorar aspectos do problema, ou quaisquer padrões que você possa ter identificado, para concentrar-se nos aspectos necessários do problema ou da solução potencial.

Refleta sobre...

- Perguntar se algumas partes do problema são menos relevantes para um resultado bem-sucedido. Alguma é supérflua? Existe algum aspecto que não levamos em consideração no início?
- Identificar situações em que considerar componentes desnecessários ou muitos elementos de um problema pode levar a tentar fazer demais, correndo o risco de comprometer uma solução eficaz.

Implementação de uma Solução



ABSTRAÇÃO

Mapas e Rotas (GPS)

Cenário: Um app de GPS precisa mostrar o caminho entre dois pontos.

Abstração aplicada: O mapa ignora detalhes como árvores, postes ou a cor das casas, e mostra só ruas, avenidas e pontos de referência relevantes.

- ✓ Essencial: Localização e conexões de vias.
- ✗ Ignorado: Detalhes visuais do ambiente.

Modelos de Machine Learning

Cenário: Criar um modelo para prever preços de casas.

Abstração aplicada: Considerar apenas variáveis como metragem, número de quartos e localização, ignorando detalhes como a cor da parede ou o modelo da geladeira.

- ✓ Essencial: Características que impactam o preço.
- ✗ Ignorado: Itens irrelevantes ou ruído nos dados.

Exemplo

Eu preciso de um bolo para a festa de aniversário da minha avó.



EU PRECISO DE UM BOLO PARA A FESTA DE ANIVERSÁRIO DA MINHA AVÓ

- Boa direção para o projeto, mas ainda em um nível muito alto para desenvolver uma solução específica. (Star Trek)
- É necessário obter mais informações sobre o bolo para decompor o problema em alguns subproblemas:
 - Que tipo de bolo ela gosta? Sabor? Estilo?
 - Ela tem alguma alergia?
 - Ela possui algum problema de saúde?
 - Alguma escrita no bolo? Decorações? Velas?

EU PRECISO DE UM BOLO PARA A FESTA DE ANIVERSÁRIO DA MINHA AVÓ

- É necessário obter mais informações sobre o contexto também:
 - Quantas pessoas vão participar? Qual deve ser o tamanho do bolo?
 - Onde será a festa — em local fechado ou ao ar livre? Clima quente ou frio?
 - Qual é a data/hora da festa? Isso nos dá um prazo para a montagem do bolo.

EU PRECISO DE UM BOLO PARA A FESTA DE ANIVERSÁRIO DA MINHA AVÓ

Após fazer algumas pesquisas, temos mais dados para nos ajudar a chegar a uma solução potencial mais específica:

- **Sabor e estilo?** Ela gosta de chocolate e frutas.
- **Alergias?** Nozes.
- **Problemas de saúde?** Nenhum.
- **Apresentação?** Um simples “Feliz Aniversário” com decorações simples, flores, bico de confeitaria e uma base. Sem velas.
- **Contexto da festa:** Trinta pessoas em local fechado na comunidade de aposentados dela. Dia 4 de maio às 16h.
- **Orçamento:** \$100.

“Preciso de um bolo para a festa de aniversário da minha avó.”

torna-se...

Preciso de um bolo de aniversário para a festa da minha avó. O bolo deve incluir chocolate e frutas, ter uma cobertura simples e não deve conter nozes. O bolo deve incluir algumas flores e detalhes orativos e dizer “Feliz Aniversário!” ...



Preciso de um bolo para a festa de aniversário da minha avó.

O bolo deve incluir chocolate e frutas.

Apresentação: algumas flores e detalhes decorativos e dizer velas

Prazo: 3 de maio
Orçamento: US\$ 100

O glacê deve ser simples, sem nozes

O bolo deve ser transportado até a casa de aposentadoria dela em 4 de maio para a festa interna às 16 horas.

Construir uma base para o bolo com algumas fotos.

Preciso de um bolo para a festa de aniversário da minha avó.

O bolo deve incluir chocolate e frutas.

Apresentação: algumas flores e detalhes decorativos e dizer velas

Prazo: 3 de maio
Orçamento: US\$ 100

O glacê deve ser simples, sem nozes

Construir uma base para o bolo com algumas fotos.

O bolo deve ser transportado até a casa de aposentadoria dela em 4 de maio para a festa interna às 16 horas.

Preciso de um bolo para a festa de aniversário da minha avó.

O bolo deve incluir chocolate e frutas.

Apresentação: algumas flores e detalhes decorativos e dizer velas

Prazo: 3 de maio
Orçamento: US\$ 100

O glacê deve ser simples, sem nozes

O bolo deve ser transportado até a casa de aposentadoria dela em 4 de maio para a festa interna às 16 horas.

~~Construir uma base para o bolo com algumas fotos.~~

Preciso de um bolo para a festa de aniversário da minha avó.

O bolo deve incluir chocolate e frutas.

Apresentação:
algumas flores e detalhes decorativos e dizer velas

Prazo: 3 de maio
Orçamento: US\$ 100

O glacê deve ser simples, sem nozes



rtado
ria dela
festa
s.

Indo para a Solução do problema...

EU PRECISO DE UM BOLO PARA A FESTA DE ANIVERSÁRIO DA MINHA AVÓ



Algoritmo do Bolo



Algoritmo do Glacê



Algoritmo da Apresentação



Algoritmo do Transporte



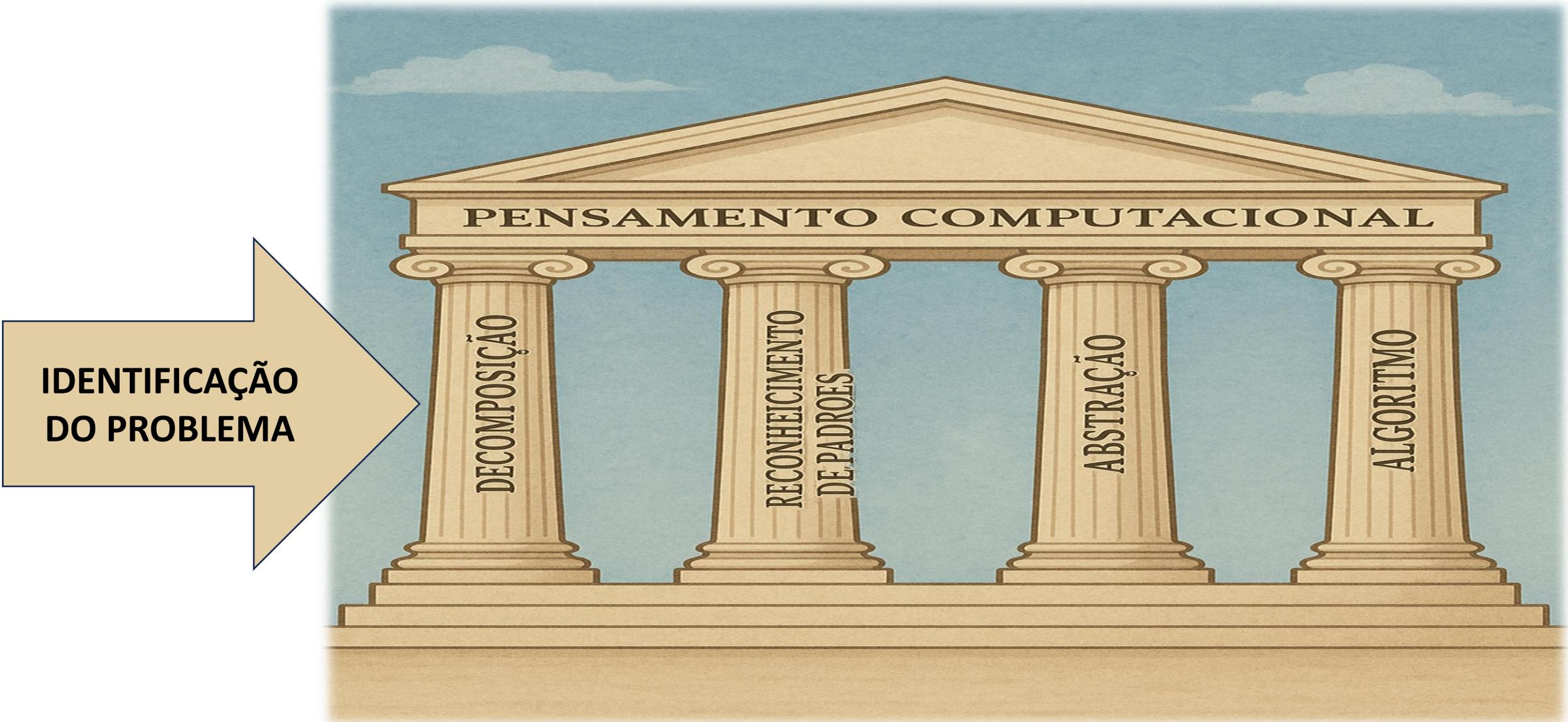
3 de Maio

4 de Maio

ORÇAMENTO: \$ 100

Agora, nós podemos pensar sobre como tornar esses "algoritmos" em um conjunto de instruções mais específicas (ou programas) para que algum "agente computacional" possa realizar...

Pensamento Computacional...



IDENTIFICAÇÃO
DO PROBLEMA

DECOMPOSIÇÃO

RECONHECIMENTO
DE PADRÕES

ABSTRAÇÃO

ALGORITMO

Pensamento Computacional...

Identificação do problema: Um computador pode ajudar a resolver meu problema?
Este problema é adequado para um computador?

Decomposição: Eu preciso dividir meu problema em partes menores para ajudar a resolvê-lo? Como as soluções para as partes menores se encaixam para resolver o problema principal?

Reconhecimento de padrões: Eu já vi abordagens de resolução de problemas no passado que podem me ajudar a resolver um dos problemas ou subproblemas que tenho aqui?

Abstração: Existem aspectos do seu problema que não são necessariamente necessários agora para resolvê-lo?

Essas etapas podem parecer lineares, mas na realidade são realizadas de maneira iterativa e não linear. E não existe uma única maneira de fazer isso!

Essas práticas representam “coisas para pensar” enquanto você trabalha para desenvolver um plano para resolver o problema de forma que depois possa começar a escrever o programa de computador.

Vamos para a prática...



Atividade: Robô Humano

- Tarefa: **Desenhar uma casa no quadro.**
- Robô: o Professor (ou algum aluno voluntário)
- Os estudantes devem dar os comandos, passo a passo, para o “robô” realizar a tarefa.



Atividade: Robô Humano



- Tarefa: **Desenhar uma casa no quadro.**
- Robô: o Professor (ou algum aluno voluntário)
- Os estudantes devem dar os comandos, passo a passo, para o “robô” realizar a tarefa.

- **Objetivo:** Exercitar **decomposição** (dividir a tarefa em passos menores) e destacar a importância de **algoritmos precisos**.
- Essa dinâmica ilustra de forma divertida como um algoritmo deve ser detalhado e ordenado para funcionar – assim como *escrever instruções para montar um sanduíche, detalhando cada passo* corretamente.
- Os alunos percebem na prática que instruções vagas levam a erros, reforçando o pensamento lógico.

Atividade: Robô Humano



- *“Que **problema** estávamos tentando resolver com o robô?”*
(Identificação do problema)
- *“Como tivemos que **decompor** a tarefa?”*
(Listar os passos encontrados)
- *“Alguém percebeu algum **padrão** na forma de dar instruções, talvez parecido com a receita do bolo?”*
(Reconhecimento de padrões, ex.: ambas as tarefas tinham sequência de fazer o desenho, executar passos em ordem)
- *“Quais detalhes tivemos que ajustar ou ignorar para o robô entender?”*
(Abstração – focar no essencial e remover ambiguidades irrelevantes).

Atividade 2 – Vamos sacar um dinheiro?

Problema: *Determinar o que um caixa eletrônico deve fazer quando alguém saca dinheiro*



Passo 1: (Think): *Identifique o problema em suas próprias palavras e pense em como resolver, passo a passo,.*

Anote brevemente as etapas que imaginou (uma sequencia de passos).

Atividade 2 – Vamos sacar um dinheiro?

Problema: *Determinar o que um caixa eletrônico deve fazer quando alguém saca dinheiro*



Passo 2: (Pair) *Forme uma dupla com seu(sua) colega ao lado. Troquem ideias. Cada dupla compara as soluções pensadas, refinando a sequência de passos...*

Cheguem a uma sequência de passos consensual.

Atividade 2 – Vamos sacar um dinheiro?

Problema: *Determinar o que um caixa eletrônico deve fazer quando alguém saca dinheiro*

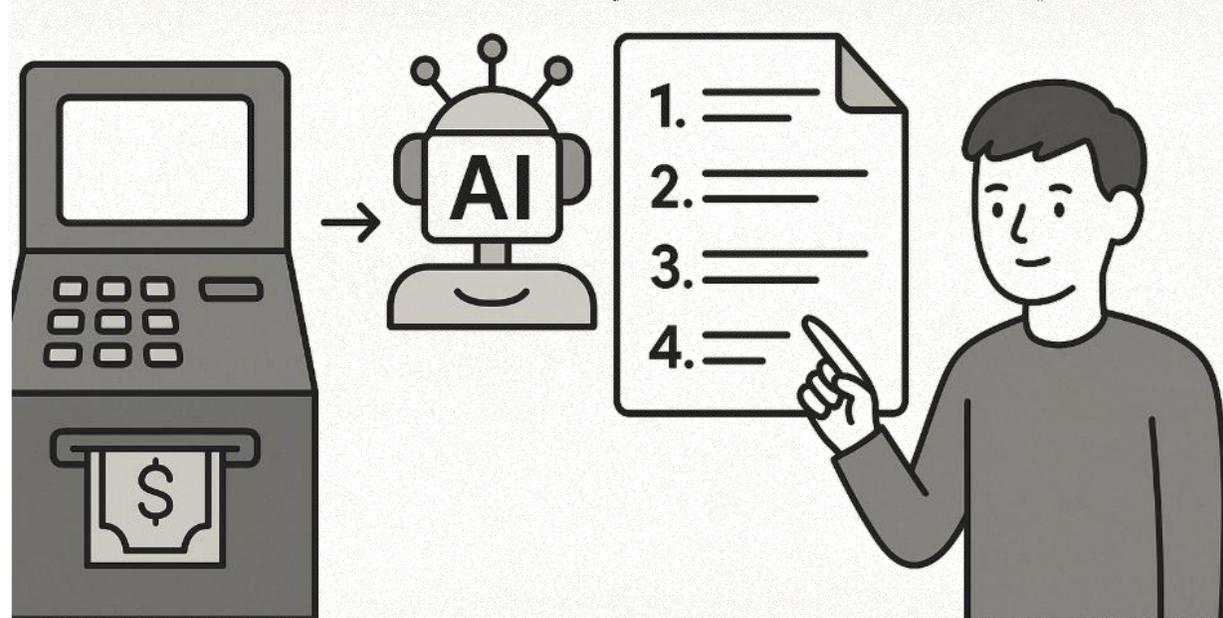


Passo 3: (Share) *Duas duplas (voluntárias) apresentam a solução (como resolveram o problema)*

Solução anotada no quadro.

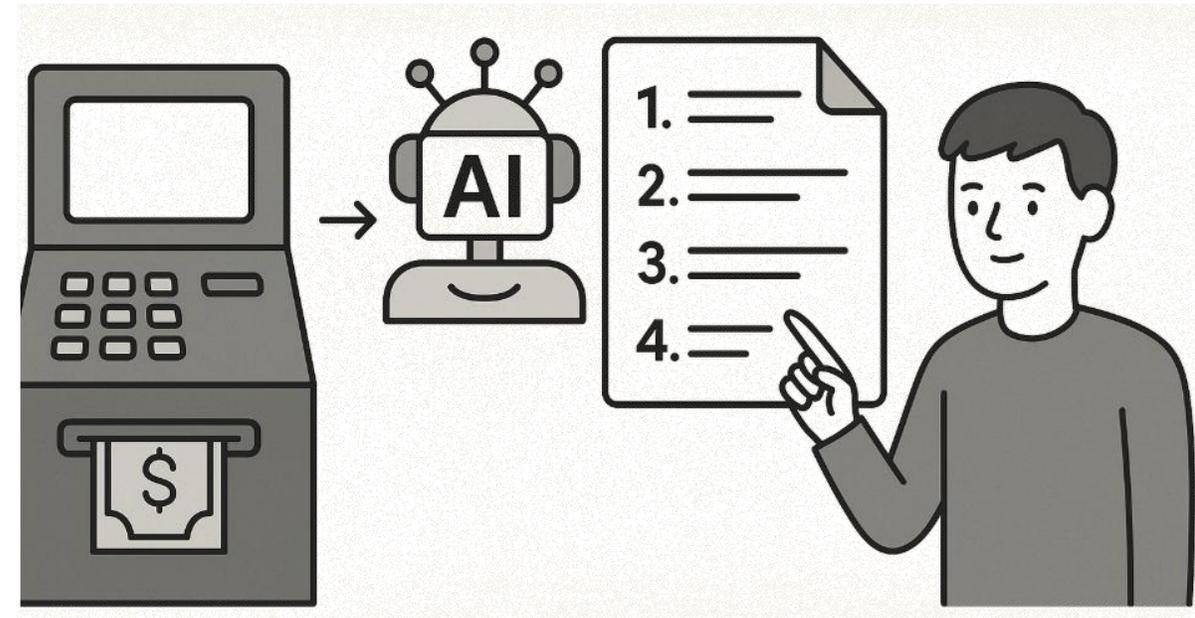
Atividade 2 (c/ IA) – Vamos sacar um dinheiro?

- Problema: ***Determinar o que um caixa eletrônico deve fazer quando alguém saca dinheiro***
- Cada dupla agora escolhe uma ferramenta de IA (ChatGPT, Gemini ou DeepSeek).
- Vamos perguntar a elas quais seriam essa sequencia de passos.
- Acrescente: ***“Como eu resolveria tal problema em pseudocódigo?”***



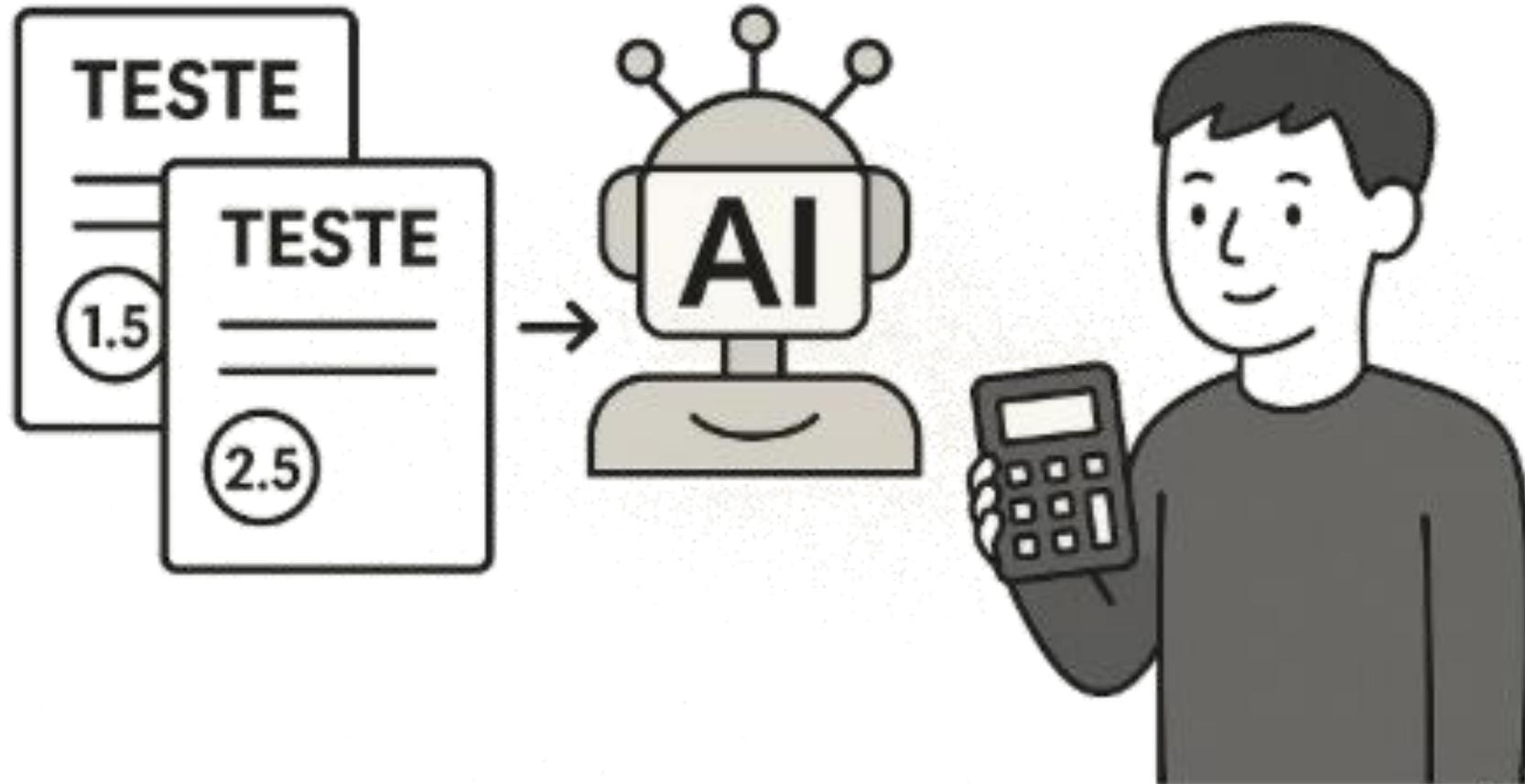
Atividade 2 (c/ IA) – Vamos sacar um dinheiro?

- Problema: ***Determinar o que um caixa eletrônico deve fazer quando alguém saca dinheiro - “Como eu resolveria tal problema em pseudocódigo?”***
- Uso ético e eficaz de IAs generativas como assistentes no aprendizado de programação... E não como muletas... Ou recorta e cola.
- IA → segundo par de olhos!
- IA → apoio na resolução de problemas.



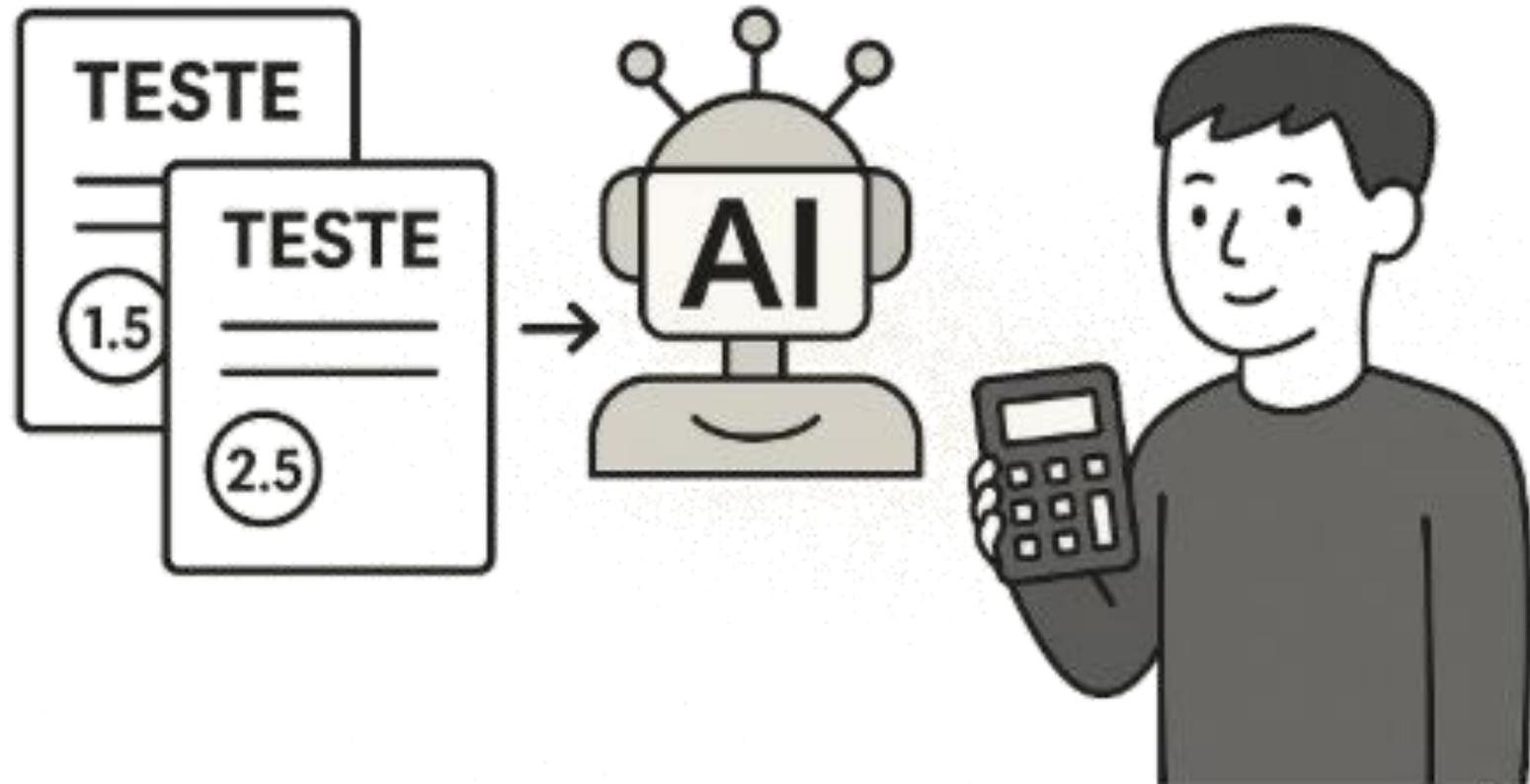
Atividade 3 – Vamos calcular a média de duas notas?

- Problema: ***Determinar os passos para calcular a média de duas notas.***
- Cada dupla : escolhe uma ferramenta de IA (ChatGPT, Gemini ou DeepSeek).
- Acrescente: ***“Como eu resolveria tal problema em pseudocódigo?”***



Atividade 3 – Vamos calcular a média de duas notas?

- Problema: ***Determinar os passos para calcular a média de duas notas.***
- AGORA: Peça a IA para transformar essa sequencia de passos, em um programa na linguagem Python, de forma mais simples possível.
- Peça para a IA EXPLICAR cada linha do código.



Próxima Aula



- Ler os conteúdos da semana 3 e ver os vídeos indicados.
- No curso online, ver a videoaula correspondente!

Boa semana e bons estudos!!