

# Refatoração do Algoritmo ACO no Problema da Mochila Binária

Acadêmicos: Matheus Gabriel, Larissa Hoffmann, Lukas Thiago e Mateus Akira

Profº: Glauco Scheffel

O QUE VEM POR AÍ

# Índice

- CONTEXTUALIZAÇÃO
- CÓDIGO ORIGINAL - PROBLEMAS IDENTIFICADOS
- PLANEJAMENTO DA REFATORAÇÃO
- ETAPAS DA REFATORAÇÃO
- RESULTADOS PÓS-REFATORAÇÃO
- CONCLUSÕES

# ACO? Refatoração?

- **Algoritmo ACO + Mochila Binária**

- ACO(Ant Colony Optimization) é uma metaheurística inspirada em comportamento de formigas.
- Usado para encontrar soluções aproximadas no problema da mochila binária.

- **Objetivo do Trabalho**

- Aplicar refatoração para melhorar estrutura e qualidade do código.
- Enfatizar boas práticas: modularização, encapsulamento e testes.

# Código Original - Problemas Identificados

- Alto acoplamento e funções misturadas
- Ausência de testes unitários automatizados
- Lógica repetida prejudicando manutenibilidade
- Legibilidade e clareza comprometidas

# Planejamento da Refatoração

## Problemas a resolver:

- Acoplamento
- repetição
- falta de testes

## Técnicas Utilizadas

- Encapsulamento
- single responsibility
- modularização

## Implementação de Testes

- Uso da biblioteca unittest para cobertura de testes

# Melhorias Aplicadas

## Classe ACO

- Encapsulamento da lógica em uma classe dedicada.

## Parâmetros Globais

- Separação clara para facilitar ajustes e reutilização.

## Método Avaliar

- Clarificação e organização da avaliação da solução.

## Testes Unitários

- Implementação com unittest para garantir estabilidade.

# Resultados Pós-Refatoração

## Testes Passando

- Confirmação da funcionalidade após alterações.

## Organização

- Código modularizado e com responsabilidade bem definida.

## Facilidade de Manutenção

- Código mais legível, simples para adaptação futura.

## Reuso Aprimorado

- Estrutura pronta para estender soluções relacionadas.