



CONTEXTO NARRATIVO

Ano 10 ABY (Após a Batalha de Yavin)

A galáxia vive um momento decisivo. A Nova República enfrenta uma nova ameaça: o Almirante Thraxx, um líder imperial renegado, está formando uma frota secreta nas Regiões Desconhecidas.

A Aliança Rebelde precisa agir — e rápido! Para isso, estão convocando jovens recrutas para programar um sistema de comando que organize as forças da Rebelião: naves, pilotos e missões.

Você e sua equipe fazem parte de uma célula rebelde. Sua missão é desenvolver um sistema funcional que permita planejar e executar missões estratégicas contra o Império.

O futuro da galáxia depende da sua lógica e do seu código.



OBJETIVO DA MISSÃO

Criar um sistema de gerenciamento de missões da Aliança Rebelde, utilizando Programação Orientada a Objetos e estruturas de dados em Python. O sistema deve permitir:

- Cadastrar naves e pilotos
- Criar missões
- Alocar pilotos e naves em missões
- Controlar disponibilidade dos recursos
- Mostrar o status das missões

ENTIDADES DO SISTEMA

- Classe: Nave
 - Atributos: nome, modelo, capacidade, disponivel
 - Métodos: mostrar informações da nave
- Classe: Piloto
 - Atributos: nome, patente, naves, em_missao
 - Métodos: adicionar nave, mostrar status
- Classe: Missao
 - Atributos: nome, destino, objetivo, pilotos, naves
 - Métodos: adicionar piloto, adicionar nave, iniciar missão, encerrar missão

REGRAS DA MISSÃO

- - Cada missão precisa de pelo menos 2 pilotos e 1 nave para ser iniciada.
- - Um piloto não pode participar de mais de uma missão por vez.
- - Naves só podem ser alocadas se estiverem disponíveis.
- - O sistema deve mostrar:
 - Lista de pilotos e naves disponíveis
 - Lista de missões em andamento
 - Lista de missões finalizadas

CARGOS DA EQUIPE REBELDE

Cada grupo é uma base rebelde que deve se organizar em equipe. Cada aluno assume um papel crucial na resistência. Todos devem colaborar para o sucesso da missão!

ENGENHEIRO REBELDE (dados): Cria os objetos de nave e piloto. Responsável pelas estruturas de listas simulando o 'banco de dados' da base.

PROGRAMADOR JEDI (classes): Responsável por escrever as classes principais e garantir que os métodos funcionem.

COMANDANTE DE MISSÃO (lógica): Cuida das regras de alocação, validações e lógica do início e fim das missões.

ANALISTA GALÁCTICO (relatórios): Desenvolve funções que mostram status das missões, recursos disponíveis e imprime relatórios.

GUARDIÃO DO CÓDIGO (testes): Revisa e testa o sistema, garantindo que tudo funcione corretamente e que todos participem.

DESAFIO EXTRA - ALERTA DE ESPIÃO

Um dos pilotos pode ser um infiltrado do Império! Crie uma função que analisa o histórico de missões e tente identificar possíveis sabotagens com base em:

- Participações repetidas em missões fracassadas
- Uso recorrente de naves danificadas
- Comportamentos suspeitos (dados simulados pelo grupo)