

Projeto de Programação II – Gestão de Entregas / Logística

Disciplina: Programação II

Professor: Welison de Brito dos Santos Batista

Turma: 3º Ano

Valor: 20 pontos

Objetivo do Projeto

Criar um sistema simples de gestão de entregas usando Programação Orientada a Objetos em Python, com persistência de dados em arquivos JSON. O sistema deve simular o funcionamento básico de uma transportadora: cadastro de motoristas e veículos, registro de entregas e atualização do status.

Funcionalidades Principais

O programa será executado no terminal com um menu, por exemplo:

1. Cadastrar motorista
2. Cadastrar veículo
3. Registrar entrega
4. Atualizar status da entrega
5. Listar entregas pendentes
6. Listar todas as entregas
0. Sair

Estrutura de Classes

1. Motorista

Atributos: id, nome, cnh.

Métodos:

`__str__`: retorna informações resumidas.

Métodos estáticos de validação (ex: validar número da CNH).

2. Veiculo

Atributos: id, placa, modelo.

Métodos:

__str__: retorna informações resumidas.

Método estático para validar placa.

3. Entrega

Atributos: id, descricao, motorista, veiculo, status.

Métodos:

atualizar_status(novo_status).

__str__: retorna informações resumidas.

4. SistemaEntrega

Responsável por gerenciar tudo: listas de motoristas, veículos e entregas.

Métodos:

cadastrar_motorista.

cadastrar_veiculo.

registrar_entrega.

atualizar_status_entrega.

listar_entregas.

Persistência em JSON (salvar e carregar dados).

Persistência de Dados (JSON)

Todos os dados (motoristas, veículos e entregas) serão salvos em arquivos .json.

Exemplo:

motoristas.json

veiculos.json

entregas.json

O programa deve carregar os arquivos ao iniciar e atualizar ao salvar.

Exceptions (Exceções)

O sistema deve prever erros comuns, como:

Tentar cadastrar um motorista com CNH inválida.

Tentar registrar uma entrega sem motoristas ou veículos cadastrados.

Tentar atualizar status de uma entrega inexistente.

Métodos Estáticos

Alguns exemplos de métodos estáticos úteis:

Validar CNH do motorista.

Validar formato de placa de veículo.

Gerar IDs automáticos para novos cadastros.

Cada método estático deve ter docstring explicando sua função.

✅ Critérios que o projeto atende

POO → uso de classes (Motorista, Veiculo, Entrega, SistemaEntrega).

JSON → persistência de dados.

Exceptions → tratamento de erros.

Métodos estáticos → validações e geração de IDs.

Organização → classes em arquivos separados, main.py só com o menu.

💡 Possíveis Extensões Criativas

Relatório de entregas por motorista.

Relatório de entregas concluídas em determinado período.

Calcular custo estimado da entrega (baseado na distância, fictícia).