

CURSO: CIÊNCIA DA COMPUTAÇÃO

DISCIPLINA: PROGRAMAÇÃO ORIENTADA A OBJETOS

PROFESSOR: KARLOS OLIVEIRA

O sistema de logística de transporte terrestre de mercadorias é uma solução desenvolvida para atender às necessidades de empresas que gerenciam operações de transporte, armazenamento e entrega de cargas. Com a crescente demanda por agilidade e eficiência no setor logístico, tornou-se importante contar com sistemas tecnológicos que permitam controlar de forma precisa os veículos, motoristas, cargas, rotas e manutenção da frota.

O principal objetivo do sistema é automatizar e otimizar o gerenciamento das operações logísticas terrestres, desde o cadastro de veículos e motoristas até o monitoramento de viagens e geração de relatórios. Através da implementação de conceitos avançados de Programação Orientada a Objetos (POO), o sistema busca proporcionar flexibilidade, escalabilidade e manutenção eficiente.

A logística é um fator estratégico para empresas que desejam reduzir custos operacionais e garantir a entrega pontual das mercadorias. Um sistema robusto como este facilita o acompanhamento em tempo real das operações, assegura a segurança das cargas e mantém o controle sobre a frota e seus condutores.

A necessidade de um sistema desse porte é evidente em um cenário onde erros humanos podem comprometer prazos e impactar diretamente na satisfação dos clientes. Com funcionalidades modernas e aplicação rigorosa de princípios da Orientação a Objetos, este sistema permite:

- **Melhor gerenciamento de frotas:** Controle completo sobre veículos e suas manutenções.
- **Organização das viagens:** Registro eficiente de rotas, motoristas e cargas transportadas.
- **Análise e tomada de decisão:** Geração de relatórios gerenciais para otimização das operações.
- **Segurança das informações:** Encapsulamento e abstração garantem a proteção dos dados críticos.

Dessa forma, sua equipe foi contratada para desenvolver esse sistema utilizando os conceitos visto na disciplina de Programação Orientada a Objetos. Assim, as Histórias de Usuário abaixo devem ser seguidas para chegar ao desenvolvimento completo do sistema.

HISTÓRIAS DE USUÁRIO – SISTEMA DE LOGÍSTICA DE TRANSPORTE TERRESTRE DE MERCADORIAS

1. Cadastro de Veículos (Classe: Veículo e Herança em Classes Filhas)

Como administrador do sistema,

Quero cadastrar diferentes tipos de veículos (caminhões leves, pesados, vans)

Para que possa organizar a frota disponível para transporte.

- **Critérios de Aceite:**

- Deve haver uma classe base Veiculo com atributos como placa, capacidade de carga e tipo de combustível.
- As classes filhas CaminhaoLeve, CaminhaoPesado e Van devem herdar de Veiculo.
- Os métodos devem ser sobrepostos para calcular o custo de manutenção com base no tipo do veículo.

2. Cadastro de Motoristas (Classe: Motorista)

Como administrador,

Quero cadastrar motoristas vinculados aos veículos,

Para que possa organizar as viagens.

- **Critérios de Aceite:**

- A classe Motorista deve conter atributos como nome, número da CNH, e experiência.
- Deve haver encapsulamento com métodos get e set para os atributos.
- O cadastro deve validar se a CNH está em dia.

3. Registro de Viagens (Classe: Viagem)

Como operador logístico,

Quero registrar uma viagem com informações do veículo, motorista, rota e carga,

Para controlar o transporte das mercadorias.

- **Critérios de Aceite:**

- A classe Viagem deve conter informações sobre origem, destino, data de saída e carga transportada.
- Construtores devem ser utilizados para inicializar as viagens.

4. Gerenciamento de Cargas (Classe: Carga)

Como operador logístico,

Quero cadastrar as mercadorias transportadas com peso e tipo,

Para controlar a carga embarcada nos veículos.

- **Critérios de Aceite:**

- A classe Carga deve possuir atributos como peso, tipo de carga e valor.
- Deve haver polimorfismo para calcular o valor do seguro com base no tipo de carga (CargaPerigosa, CargaFrágil).

5. Interface de Pagamento (Interface: Pagamento)

Como gestor financeiro,

Quero que o sistema permita o cálculo de valores de transporte,

Para faturamento correto das operações.

- **Critérios de Aceite:**
 - A interface Pagamento deve definir métodos para calcular valor da viagem e taxas adicionais.
 - Deve ser implementada em classes concretas como PagamentoPorPeso e PagamentoPorDistancia.

6. Interface de Relatórios (Interface: Relatorio)

Como gerente operacional,

Quero gerar relatórios das viagens realizadas e status dos veículos,

Para análise de desempenho.

- **Critérios de Aceite:**
 - A interface Relatorio deve definir métodos para exibir informações de viagens e manutenção de veículos.

7. Cadastro de Rota (Classe: Rota)

Como operador logístico,

Quero cadastrar rotas com distância e condição da estrada,

Para otimizar as operações de transporte.

- **Critérios de Aceite:**
 - A classe Rota deve permitir o cadastro de distância, pontos de parada e condições da estrada.
 - Deve ser possível sobrescrever um método para calcular o tempo estimado com base na condição da estrada.

8. Gerenciamento de Manutenção (Classe: Manutencao)

Como gestor de frota,

Quero registrar manutenções realizadas nos veículos,

Para garantir a segurança das operações.

- **Critérios de Aceite:**
 - A classe Manutencao deve registrar data, tipo de manutenção e custo.

9. Interface de Notificação (Classe: Notificacao)

Como operador logístico,

Quero receber notificações sobre veículos em manutenção,

Para organizar as operações de transporte.

- **Critérios de Aceite:**
 - Deve haver polimorfismo para notificar via e-mail ou SMS.

10. Diagrama de Classes

Como desenvolvedor,

Quero criar um diagrama de classes detalhado,

Para representar a estrutura do sistema.

- **Critérios de Aceite:**
 - O diagrama deve conter todas as classes, interfaces, relacionamentos, atributos e métodos.

Requisitos Técnicos:

- Implementação com os 4 pilares da POO: abstração, encapsulamento, herança e polimorfismo.
- Uso de construtores, métodos get e set.
- Implementação de duas interfaces.
- Sobrecarga e sobreposição de métodos.
- Além das classes propostas aqui, o sistema deve possuir no mínimo **13 classes e/ou interfaces**.

Instruções Gerais:

- **Equipe:** O trabalho deverá ser realizado em grupos de 3 pessoas. Cada parte do trabalho deve indicar qual aluno foi o principal responsável, porém, todos os membros devem colaborar em todas as partes.
- **Apresentação:** Em um formato de Pitch, cada grupo deverá explicar o código, demonstrar o sistema funcionando e mostrar a representação gráfica em UML.

Entrega:

- Apresentação das funcionalidades em sala.
- Diagrama UML representando as classes e os relacionamentos.
- Indicação na apresentação de qual membro da equipe foi o principal responsável por cada parte.
- Data das apresentações: 17 e 18 de Fevereiro