

## Recuperação Ana Trabalho Java

A classe principal é EmpresaTurismo, onde o método main é responsável por executar o programa. Dentro do main, três listas são criadas para armazenar objetos das classes Veiculos, Abastecimento e Gasto. Essas listas servem para gerenciar os dados dos veículos, os abastecimentos realizados e os gastos associados à frota.

### 1º Parte: EmpresaTurismo

No início do código pensei em fazer o Menu interativo das opções para ser vistas durante a ação do código onde foi colocado as opções de : Cadastrar veículo,Cadastrar gasto,Calcular consumo do combustível,visualizar relatórios,sair e opções Inválidas .

```
List<Veiculos> veiculo = new ArrayList<>();
List<Abastecimento> Abastecimento = new ArrayList<>();
List<Gasto> gastos = new ArrayList<>();

while(true) {

    String opcao = JOptionPane.showInputDialog("Selecione uma opção:\n1 - Cadastrar veículo\n2 - Cadastrar gasto\n3 - Calcular consumo de combustível\n4 - Visualizar relatórios\n5 - Sair\n6 ");
    switch (opcao) {
        case "1":
            CadastrarVeiculo();
            break;

        case "2":
            CadastrarAbastecimento();
            break;

        case "3":
            CadastrarGasto();
            break;

        case "4":
            CalcularConsumoCombustivel();
            break;

        case "5":
            VisualizarRelatorio();
            break;

        case "6":
            System.exit(0);

        default:
            JOptionPane.showMessageDialog(null, "Opção Inválida");
    }
}
```

No início decidir usar o List para uma coleção indexada de objetos. Onde são baseados em zero. Foi criada uma list para veículos,Abastecimento e gastos. Onde depois de fazer o list foi usado um switch case para substituir o if e else,onde foi tendo as opção a ser mostrada no menu.

#### Opção 1 - Cadastrar veículos

A opção "1" deve permitir ao usuário cadastrar um novo veículo. O método CadastrarVeículo seria responsável por coletar as informações do veículo e adicioná-las à lista de veículos.

#### Opção 2 - Cadastrar gasto

A opção "2" seria usada para registrar um gasto associado a um veículo. Isso pode incluir manutenção, reparos, entre outros.

#### Opção 3 - Calcular consumo de combustível

A opção "3" deve calcular o consumo de combustível dos veículos com base nos abastecimentos registrados. O método `CalcularConsumoCombustivel` seria responsável por realizar esse cálculo.

#### Opção 4 - Visualizar relatórios

A opção "4" deve gerar e exibir relatórios de despesas, consumo de combustível, e outros dados relevantes para o consumidor. O método `VisualizarRelatório` é o que deverá ser utilizado para isso exibir todos os gastos durante o mês de uso.

#### Opção 5 - Sair

A opção "5" encerra o programa.

#### Opção 6 - Opções Inválidas

Caso o usuário insira uma opção que não esteja entre as listadas (1 a 6), o programa exibe uma mensagem de erro informando que a opção é inválida.

## 2º Parte: Cadastrar veículo

Nessa parte de cadastrar o veículo foi feito errado, mas vou explicar na minha visão.

```
veiculoset.Marca(JOptionPane.showInputDialog("Marca do veículo:"));
veiculoset.Modelo(JOptionPane.showInputDialog("Modelo do veículo:"));
veiculoset.AnoFabricacao(JOptionPane.showInputDialog("Ano de fabricação:"));
veiculoset.AnoModelo(JOptionPane.showInputDialog("Ano do modelo do veículo:"));
veiculoset.Motorizado(JOptionPane.showInputDialog("Motorização do veículo:"));
veiculoset.CapacidadeTanque(JOptionPane.showInputDialog("Capacidade do tanque (em litros:)"));
veiculoset.Combustivel(JOptionPane.showInputDialog("Tipo de combustivel do veículo:"));
veiculoset.Cor(JOptionPane.showInputDialog("Cor do veículo:"));
veiculoset.Placa(JOptionPane.showInputDialog("Placa do veículo:"));
veiculoset.Renavam(JOptionPane.showInputDialog("Renavam do veículo:"));

veiculo.add(veiculo);
JOptionPane.showInputDialog("Veículo cadastrado com sucesso!");
```

No início decidir usar o Set para armazenar um conjunto de elementos únicos e não interfere na ordem em que esses elementos são armazenados.

Onde decidir usar como Veiculo.set.marca(JOptionPane.showInputDialog) mas a frase de exibição onde mostraria cada opção de especificação do veículo. Mas acabei fazendo errado, onde o certo teria que ser usar o private String marca para nomes e private Double para números e usar o método getter e setters.

### 3º Parte: Cadastrar Abastecimento

```
private static void CadastrarAbastecimento(){  
  
    Object veiculo = null;{  
        JOptionPane.showMessageDialog(null, "Não há veículos cadastrados. Cadastre um veículo primeiro.");  
        return;  
    }  
  
    Abastecimento abastecimento = new Abastecimento();  
    Abastecimento.setvalorAbastecimento(Double.parseDouble(JOptionPane.showInputDialog("Valor do abasecimento:")));  
  
    Abastecimento.setQuantidadeCombustivel(Double.parseDouble(JOptionPane.showInputDialog("Quantidade de combustivel abastecida (em litros):")));  
    abastecimento.setQuilometragemAtual(Double.parseDouble(JOptionPane.showInputDialog("Quilometragem atual do veículo")));  
  
    abastecimento.add(abastecimento);  
    JOptionPane.showMessageDialog(null, "Abastecimento registrado com sucesso!");  
}
```

No início do código decidir já usar o JOptionPane.showMessageDialog Para exibir uma mensagem dizendo que “ não há veículos Cadastrados para o abastecimento”

```
Abastecimento abastecimento = new Abastecimento();  
Abastecimento.setvalorAbastecimento(Double.parseDouble(JOptionPane.showInputDialog("Valor do abasecimento:")));  
  
Abastecimento.setQuantidadeCombustivel(Double.parseDouble(JOptionPane.showInputDialog("Quantidade de combustivel abastecida (em litros):")));  
abastecimento.setQuilometragemAtual(Double.parseDouble(JOptionPane.showInputDialog("Quilometragem atual do veículo")));
```

Para abastecer decidir usar o :

- Abastecimento.setvalorAbastecimento
- Abastecimento.setQuantidadeCombustivel
- Abastecimento.setQuilometragemAtual

O double foi usado para ação do abastecimento :

Double.parseDouble(JOptionPane.showInputDialog) para exibir a mensagem na interface .

Depois foi usado o “abastecimento.add(abastecimento);

JOptionPane.showMessageDialog(null, "Abastecimento registrado com sucesso!");

Para mostrar na interface quando o cadastro for Concluído da maneira correta.

### 4º Parte: Cadastrar consumo do combustível

Já de início especifiquei que seria necessário de no máximo dois abastecimentos completos para fazer o cálculo do consumo

```

if(Abastecimento.size()<2){
    JOptionPane.showMessageDialog(null, "É necessário ter pelo menos dois abastecimentos completos para calcular o consumo.");
    return;
}

```

```

if(Abastecimento.size()<2){
    JOptionPane.showMessageDialog(null, "É necessário ter pelo menos dois
abastecimentos completos para calcular o consumo.");
    return;
}

```

```

}
Double quilometrosTotais = Abastecimento.get(Abastecimento.size() - 1).getQuilometragemAtual() - Abastecimento.get(0).getQuilometragemAtual();
int litrosConsumidos = 0;

for(int i = 1; i < Abastecimento.size(); i++) {

```

Aqui decidir usar o double é o int para fazer o cadastro do quilómetros total e os litros consumidos.

Na última parte do código, usei o "double consumoMedio = quilometrosTotais/litrosConsumidos;

```

JOptionPane.showMessageDialog(null, "O consumo médio é de:" + consumoMedio +
"km/l");

```

Onde mostra o consumo médio ele vai ser igual quilometros totais Dividido por litros consumidos pra ter uma noção de gasto durante a viagem.

### Cadastrar gasto

No início decidir usar o if e o isEmpty "if(Veiculos.isEmpty()){"

```

JOptionPane.showMessageDialog(null, "Não há veículos cadastrados. Cadastre um
veículo primeiro.");
return;

```

Já na segunda parte do código, usei o "gasto.setCategoria e o gasto.setValor " para adicionar qual foi o gasto e o valor.

Depois foi feito o gasto.add mais o JOptionPane.showMessageDialog Para aparecer a mensagem de que foi cadastrado.

```

if(Veiculos.isEmpty()){
    JOptionPane.showMessageDialog(null, "Não há veículos cadastrados. Cadastre um veículo primeiro.");
    return;
}

Gasto gasto = new Gasto();
gasto.setCategoria(JOptionPane.showInputDialog("Categoria de gasto:"));
gasto.setValor(JOptionPane.showInputDialog("Valor do gasto:"));

gasto.add(gasto);
JOptionPane.showMessageDialog(null, "Gasto registrado com sucesso!");

```

## 4º Parte: Vizualizar Relatórios

Para mostrar o relatório do veículo usei o if e o switch Case e o StringBuilder,o motivo de eu usar os StringBuilder foi para poder criar variáveis de String Modificáveis.

```
if(Veiculo.isEmpty() && abastecimento.isEmpty()) {  
  
    JOptionPane.showMessageDialog(null, "Não há  
    return;  
}  
StringBuilder relatorio = new StringBuilder();  
String opcaoRelatorio = JOptionPane.showInputDialog(  
  
    switch (opcaoRelatorio) {  
        case "1" :  

```

Onde foi usado o StringBuilder junto com o JOptionPane.showMessageDialog Para exibir que no momento não há nada cadastrado.

Já com o switch Case usei como opção de relatório cada um com o número para ser identificado na hora de cadastrar, onde foi usado o append onde foi usado cada um com sua função. Exemplo: Relatório.append(abastecimento.toString()).append("\n2") Caso a pessoa inserir uma opção que não exista, exibirá uma mensagem usando o JOptionPane.showMessageDialog " Opção de relatório inválida!".

```

switch (opcaoRelatorio) {
    case "1" :
        for(Veiculo veiculo : veiculos){
            relatorio.append(veiculo.toString()).append("\n1");
        }
        break;

    case "2" :
        for(Abastecimento abastecimento : abastecimentos){
            relatorio.append(abastecimento.toString()).append("\n2");
        }
        break;

    case "3":
        for(Gasto gasto : Gastos){
            relatorio.append(gasto.toString()).append("\n");
        }
        break;

    case "4":
        relatorio.append("Relatório de veículos:\n");
        for(Veiculo veiculo : Veiculos){
            relatorio.append(veiculo.toString()).append("\n");
        }
        relatorio.append("\nRelatório de abastecimento:\n");
        {
            Iterable<Abastecimento> Abastecimentos = null;
            for(Abastecimento abastecimento : Abastecimentos){
                relatorio.append(abastecimento.toString()).append("\n");
            }
        }

        relatorio.append("\nRelatório de gastos:\n");
        for(Gasto gasto : Gastos){
            relatorio.append(gasto.toString()).append("\n");
        }
        break;

    default:
        JOptionPane.showMessageDialog(null, "Opção de relatório inválida");
}

```

Aluna;Ana Luiza  
 Recuperação do trabalho de java