

## Diagrama de Classe UML

### Classe: **Veiculo**

- **Atributos:**
  - - modelo: String
  - - placa: String
  - - anoFabricacao: Int
  - - cor: String
  - - tipoCombustivel: String
  - - historicoManutencao: List<Manutencao>
  - - consumoCombustivel: Float
- **Métodos:**
  - + cadastrarVeiculo(modelo, placa, anoFabricacao, cor, tipoCombustivel): void
  - + atualizarVeiculo(modelo, placa, anoFabricacao, cor, tipoCombustivel): void
  - + consultarInformacoes(): Veiculo
  - + realizarBaixa(): void

### Classe: **Motorista**

- **Atributos:**
  - - nome: String
  - - CPF: String
  - - CNH: String
  - - validadeCNH: Date
  - - historicoConducao: List<Conducao>
- **Métodos:**
  - + cadastrarMotorista(nome, CPF, CNH, validadeCNH): void
  - + atualizarMotorista(nome, CPF, CNH, validadeCNH): void
  - + consultarInformacoes(): Motorista
  - + associarVeiculo(veiculo: Veiculo): void

### Classe: **Rota**

- **Atributos:**
  - - idRota: Int
  - - pontosDestino: List<String>
  - - tempoEstimado: Float
  - - monitoramentoVeiculo: GPS
  - - relatorioRotas: List<RelatorioRota>
- **Métodos:**

- + planejarRota(pontosDestino, tempoEstimado): void
- + monitorarVeiculo(gps: GPS): void
- + gerarRelatorioRota(): RelatorioRota

## Diagrama de Classes UML com Visibilidade

### 1. Classe: Veículo

- **Atributos:**
    - - id: int
    - + modelo: String
    - + placa: String
    - + ano: int
    - - consumoMedio: float
  - **Métodos:**
    - + registrarAbastecimento(abastecimento: Abastecimento): void
    - + calcularConsumoMedio(): float
    - + adicionarDocumento(documento: Documento): void
- 

### 2. Classe: Manutenção

- **Atributos:**
    - - id: int
    - + tipo: String // (preventiva, corretiva)
    - + dataAgendamento: Date
    - + dataExecucao: Date
    - + descricao: String
    - - custo: float
    - + veiculo: Veículo
  - **Métodos:**
    - + agendar(data: Date): void
    - + registrarExecucao(data: Date, custo: float, descricao: String): void
    - + gerarRelatorio(): String
- 

### 3. Classe: Abastecimento

- **Atributos:**
  - - id: int

- + data: Date
  - + quantidade: float
  - + valor: float
  - + veiculo: Veículo
  - **Métodos:**
    - + registrar(): void
- 

#### 4. Classe: Documento

- **Atributos:**
    - - id: int
    - + tipo: String // (licenciamento, seguro, etc.)
    - + dataValidade: Date
    - + arquivo: String // Caminho do arquivo
  - **Métodos:**
    - + upload(arquivo: String): void
    - + validade(): boolean
- 

#### 5. Classe: RelatorioManutencao

- **Atributos:**
    - - manutencaoList: List<Manutenção>
  - **Métodos:**
    - + gerarRelatorio(): String
- 

#### 6. Classe: RelatorioCombustivel

- **Atributos:**
  - - abastecimentoList: List<Abastecimento>
- **Métodos:**
  - + gerarRelatorio(): String

# Diagrama de Classes UML

## 1. Classe: Veículo

- **Atributos:**
    - - id: int
    - + modelo: String
    - + placa: String
    - + ano: int
    - - consumoMedio: float
    - - historicoManutencao: List<Manutencao>
    - + consumoCombustivel: float
  - **Métodos:**
    - + cadastrarVeiculo(modelo: String, placa: String, anoFabricacao: Int, cor: String, tipoCombustivel: String): void
    - + atualizarVeiculo(modelo: String, placa: String, anoFabricacao: Int, cor: String, tipoCombustivel: String): void
    - + consultarInformacoes(): Veiculo
    - + realizarBaixa(): void
    - + registrarAbastecimento(abastecimento: Abastecimento): void
    - + calcularConsumoMedio(): float
    - + adicionarDocumento(documento: Documento): void
- 

## 2. Classe: Motorista

- **Atributos:**
  - - nome: String
  - - CPF: String
  - - CNH: String
  - - validadeCNH: Date
  - - historicoConducao: List<Conducao>
- **Métodos:**
  - + cadastrarMotorista(nome: String, CPF: String, CNH: String, validadeCNH: Date): void
  - + atualizarMotorista(nome: String, CPF: String, CNH: String, validadeCNH: Date): void
  - + consultarInformacoes(): Motorista
  - + associarVeiculo(veiculo: Veiculo): void

---

### 3. Classe: Rota

- **Atributos:**

- - idRota: Int
- - pontosDestino: List<String>
- - tempoEstimado: Float
- - monitoramentoVeiculo: GPS
- - relatorioRotas: List<RelatorioRota>

- **Métodos:**

- + planejarRota(pontosDestino: List<String>, tempoEstimado: Float): void
  - + monitorarVeiculo(gps: GPS): void
  - + gerarRelatorioRota(): RelatorioRota
- 

### 4. Classe: Manutenção

- **Atributos:**

- - id: int
- + tipo: String // (preventiva, corretiva)
- + dataAgendamento: Date
- + dataExecucao: Date
- + descricao: String
- - custo: float
- + veiculo: Veículo

- **Métodos:**

- + agendar(data: Date): void
  - + registrarExecucao(data: Date, custo: float, descricao: String): void
  - + gerarRelatorio(): String
- 

### 5. Classe: Abastecimento

- **Atributos:**

- - id: int
- + data: Date
- + quantidade: float
- + valor: float
- + veiculo: Veículo

- **Métodos:**

- `+ registrar(): void`
- 

## 6. Classe: Documento

- **Atributos:**
    - `- id: int`
    - `+ tipo: String` // (licenciamento, seguro, etc.)
    - `+ dataValidade: Date`
    - `+ arquivo: String` // Caminho do arquivo
  - **Métodos:**
    - `+ upload(arquivo: String): void`
    - `+ validade(): boolean`
- 

## 7. Classe: RelatorioManutencao

- **Atributos:**
    - `- manutencaoList: List<Manutenção>`
  - **Métodos:**
    - `+ gerarRelatorio(): String`
- 

## 8. Classe: RelatorioCombustivel

- **Atributos:**
    - `- abastecimentoList: List<Abastecimento>`
  - **Métodos:**
    - `+ gerarRelatorio(): String`
- 

## Resumo das Junções

As relações entre as classes já foram mencionadas anteriormente, mas para uma visão clara:

- **Veículo 1 ---- N Manutenção**
- **Veículo 1 ---- N Abastecimento**
- **Veículo 1 ---- N Documento**
- **Motorista N ---- M Veículo**
- **Rota N ---- 1 Veículo**
- **RelatorioManutencao N ---- 1 Manutenção**
- **RelatorioCombustivel N ---- 1 Abastecimento**

Com essa estrutura, você pode visualizar claramente as classes e suas interações. Se precisar de mais detalhes ou uma representação gráfica, estou à disposição!