

# Sistema de Gerenciamento de Serviços Automotivos

## Objetivo do Sistema:

O principal objetivo do sistema é otimizar a gestão de serviços de detalhamento e funilaria automotiva, facilitando a interação entre clientes, detailers e funileiros, além de proporcionar controle administrativo eficiente sobre as operações, agendamentos e pagamentos.

## Descrição do Problema a Ser Resolvido:

Atualmente, a gestão de agendamentos, o controle de serviços realizados, o acompanhamento de clientes e o gerenciamento de profissionais (detailers e funileiros) em oficinas de detalhamento e funilaria podem ser desorganizados e ineficientes. Isso pode levar a:

**Perda de agendamentos:** Dificuldade em controlar a disponibilidade dos profissionais e evitar conflitos de horário.

**Comunicação ineficaz:** Clientes e profissionais podem ter dificuldade em se comunicar sobre os serviços e seus andamentos.

**Falta de histórico:** Dificuldade em acessar o histórico de serviços de um veículo ou cliente específico.

**Gerenciamento financeiro complicado:** Ausência de um sistema para registrar pagamentos e ajustar valores de forma centralizada.

**Dificuldade de controle administrativo:** O administrador não tem uma visão clara das operações, dificultando a tomada de decisões e a supervisão.

O sistema visa resolver esses problemas, centralizando as informações e automatizando processos-chave para garantir uma operação mais fluida e profissional.

## Perfis de Usuários Envolvidos:

O sistema será utilizado por quatro perfis principais de usuários, cada um com responsabilidades e permissões específicas:

**Admin (Administrador):** Responsável pela gestão e configuração geral do sistema. Possui controle total sobre cadastros, serviços e valores.

**Cliente:** Proprietário do veículo que busca e contrata os serviços. Realiza agendamentos e acompanha o status de seus veículos.

**Funileiro:** Profissional responsável pelos serviços de reparo na lataria do veículo. Gerencia seus próprios serviços e pode ajustar preços.

**Detailer:** Profissional responsável pelos serviços de estética e detalhamento veicular. Gerencia seus próprios serviços e pode ajustar preços.

## **Lista de Operações que o Aplicativo Deve Realizar:**

**O aplicativo deve realizar as seguintes operações:**

**Entrar no sistema:** Permite que todos os perfis de usuários acessem o sistema após autenticação.

**Sair do sistema:** Permite que todos os usuários encerrem sua sessão no sistema.

**Listar serviços:** Todos os perfis podem visualizar os serviços disponíveis ou os serviços relacionados às suas atividades.

**Criar conta:** Permite que novos clientes se registrem no sistema.

### **Operações do Admin:**

**Manter o cadastro de Detailers:** Incluir, alterar e remover informações de detailers.

**Manter o cadastro de Clientes:** Incluir, alterar e remover informações de clientes.

**Manter o cadastro de Funileiros:** Incluir, alterar e remover informações de funileiros.

**Reajustar valores dos serviços:** Ajustar os preços dos serviços oferecidos no sistema.

### **Operações do Cliente:**

**Cadastrar carro:** Registrar informações sobre seus veículos no sistema.

**Agendar serviço:** Solicitar e agendar serviços para seus veículos.

**Realizar pagamento:** Efetuar o pagamento pelos serviços contratados.

### **Operações do Funileiro:**

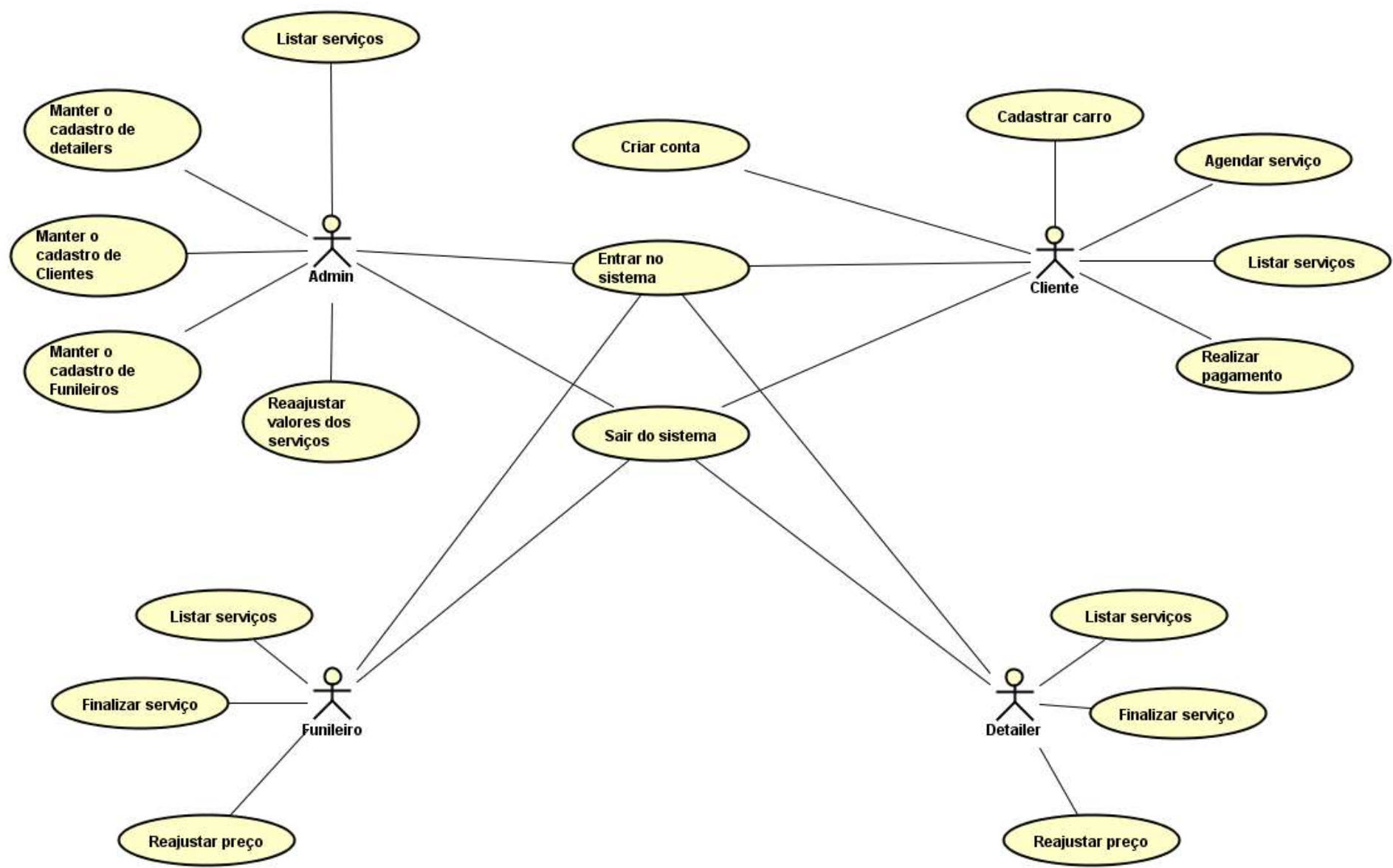
**Finalizar serviço:** Marcar um serviço como concluído após a execução.

**Reajustar preço:** Possibilidade de ajustar o preço de um serviço específico (provavelmente para orçamentos ou serviços personalizados).

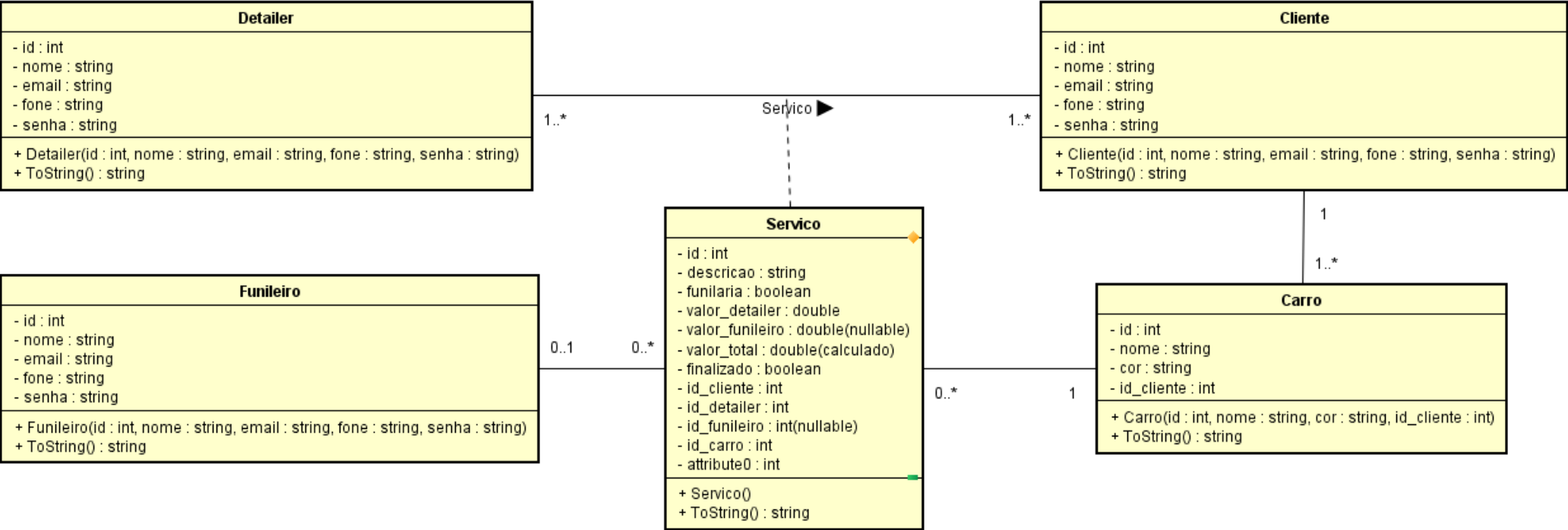
### **Operações do Detailer:**

**Finalizar serviço:** Marcar um serviço como concluído após a execução.

**Reajustar preço:** Possibilidade de ajustar o preço de um serviço específico (provavelmente para orçamentos ou serviços personalizados).



pkg



—>>

<<static>> Carros
- carros : List<Carro>
+ Inserir(obj : Carro) : void + Listar() : List<Carro> + Listar_ID(id : int) : Carro + Atualizar(obj : Carro) : void + Excluir(obj : Carro) : void + Abrir() : void + Salvar() : void

<<static>> Clientes
- clientes : List<Cliente>
+ Inserir(obj : Cliente) : void + Listar() : List<Cliente> + Listar_ID(id : int) : Cliente + Atualizar(obj : Cliente) : void + Excluir(obj : Cliente) : void + Abrir() : void + Salvar() : void

—»

<<static>> Detailers
- detailers : List<Detailer>
+ Inserir(obj : Detailer) : void + Listar() : List<Detailer> + Listar_ID(id : int) : Detailer + Atualizar(obj : Detailer) : void + Excluir(obj : Carro) : void + Abrir() : void + Salvar() : void

<<static>> Funileiros
- funileiros : List<Funileiro>
+ Inserir(obj : Funileiro) : void + Listar() : List<Funileiro> + Listar_ID(id : int) : Funileiro + Atualizar(obj : Funileiro) : void + Excluir(obj : Funileiro) : void + Abrir() : void + Salvar() : void

<<static>> Servicos
- servicos : List<Servico>
+ Inserir(obj : Servico) : void + Listar() : List<Servico> + Listar_ID(id : int) : Servico + Atualizar(obj : Servico) : void + Excluir(obj : Servico) : void + Abrir() : void + Salvar() : void