INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE – CÂMPUS PELOTAS - VISCONDE DA GRAÇA
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

AUTOCHASE: WEBSITE DE BUSCA, ANÚNCIOS E VENDAS DE VEÍCULOS DE MÉDIO E PEQUENO PORTE

Andrei Buss Farias dos Santos
Otávio da Silva Pacheco

Pelotas, 2025

Andrei Buss Farias dos Santos e Otávio da Silva Pacheco

AUTOCHASE: WEBSITE DE BUSCA, ANÚNCIOS E VENDAS DE VEÍCULOS DE MÉDIO E PEQUENO PORTE

Trabalho de Conclusão de Curso apresentado como requisito na disciplina Metodologia de Pesquisa II do curso Técnico em Desenvolvimento de Sistemas, do Instituto Federal Sul-rio-grandense Campus Pelotas - Visconde da Graça.

Orientador: Dr. Fernando Augusto Treptow Brod.

Pelotas, 2025

SUMÁRIO

1. INTRODUÇÃO 6
2. TEMA 6
3. MOTIVAÇÕES 6
4. OBJETIVOS 7
    4.1. OBJETIVO GERAL 7
    4.2. OBJETIVOS ESPECÍFICOS 7
5. ESPECIFICAÇÃO DE REQUISITOS 7
    5.1. MÉTODOS DE ESPECIFICAÇÃO DE REQUISITOS 7
    5.2. REQUISITOS FUNCIONAIS 7
    5.3. REQUISITOS NÃO FUNCIONAIS 9
6. MODELAGEM 10
    6.1. MODELO DE CASOS DE USO 10
    6.2. MODELAGEM CONCEITUAL DO BANCO DE DADOS. 11
    6.2.1. Usuário: 12
    6.2.2. Anúncio: 13
    6.2.3. Veículo: 14
    6.2.4. Modelo: 14
    6.2.5. Cor: 15
    6.2.6. Marca: 15
    6.2.7. Combustível: 15
    6.2.8. Chassi: 15
    6.2.9. Imagem: 16
    6.3. MODELAGEM LÓGICA DO BANCO DE DADOS. 16
    6.3.1. Atributos de Usuário: 17
    6.3.2. Atributos de anúncio: 18
    6.3.3. Atributos veículo: 18
    6.3.4. Atributos de Marca: 19
    6.3.5. Atributos de Modelo: 19
    6.3.6. Atributos cor: 20
    6.3.7. Atributos combustível: 20
    6.3.8. Atributos chassi: 20
    6.3.9. Atributos imagem: 21
7. TECNOLOGIAS UTILIZADAS 21
8. DESCRIÇÃO DO SISTEMA 23
    8.1. Tela Inicial 23
    8.2. Tela de Login 25
    8.3. Tela de Cadastro 27
    8.4. Tela de Edição do Usuário 29
    8.5. Tela de Compra 31
    8.6. Tela de Criação do Anúncio 33
    8.7. Tela Meus Anúncios 34
    8.8. Tela Editar Meus Anúncios 37

8.10. Tela Inicial do Painel de Administrador **42**
8.11. Tela de Edições do Painel de Administrador **51**
8.12. Estrutura de pastas do sistema **58**
8.13. Camada de configuração **58**
8.15. Apresentação e Estilo (css/, font/, img/, js/) **59**
8.16. Gerenciamento de Arquivos (uploads/) **60**
8.17. Arquivos na Raiz **60**
9. CONSIDERAÇÕES FINAIS **60**
10. REFERÊNCIAS **62**

RESUMO

O presente trabalho tem como tema o desenvolvimento do Autochase, um website voltado para busca, anúncios e vendas de veículos de pequeno e médio porte, visando atender às necessidades de compradores e vendedores de forma simples, segura e eficiente. O objetivo principal do projeto é criar uma plataforma especializada e intuitiva, que facilite negociações, permitindo o cadastro de usuários e veículos, criação e gerenciamento de anúncios, além de oferecer mecanismos avançados de busca com filtros detalhados. A metodologia empregada para a especificação de requisitos envolveu o levantamento das necessidades do usuário, definindo requisitos funcionais e não funcionais, bem como a modelagem do sistema através de casos de uso e do modelo Entidade-Relacionamento (ER). O desenvolvimento foi realizado utilizando HTML, CSS e JavaScript para o front-end, MySQL para gerenciamento do banco de dados. O sistema resultante permite que usuários realizem cadastro, login, publicação e gerenciamento de anúncios, bem como buscas complexas por veículos, com acesso seguro aos dados, respeitando normas legais e princípios éticos de privacidade. A expectativa é que o Autochase se torne uma solução competitiva no mercado de e-commerce automotivo, democratizando o acesso às vendas online e proporcionando uma experiência mais agradável e confiável para os usuários. Como considerações finais, o projeto busca não apenas atender a uma demanda existente, mas também propor inovações no setor automotivo digital, tornando a compra e venda de veículos mais acessível, segura e transparente, incentivando a competitividade e a modernização do comércio eletrônico no Brasil.

Palavras-chave: Marketplace, Automação, Segurança Digital.

1. INTRODUÇÃO

O ramo automotivo é o sexto ramo mais lucrativo do mundo, de acordo com a IBIS World, 2023. Só no Brasil, já são mais de 123 milhões de automóveis registrados, entretanto, mesmo que esse número só cresça daqui alguns anos, o poder de comercialização desses veículos encontra-se nas mãos de poucas plataformas de E-commerce e marketplace, que muitas vezes, sequer são focadas em veículos.

Encontrar o veículo ideal em um website, pode se tornar um desafio devido a tal concentração de poder que resulta em problemas, principalmente de baixa competição na área e por fim, afeta o consumidor final desses veículos. Em alguns casos, a dificuldade de acesso a informações pode acabar coagindo alguém a não realizar uma compra em um site de automóveis.

O objetivo deste trabalho **foi** desenvolver um marketplace 100% focado em veículos de leve e médio porte, competitivo e acessível para o usuário que desejava vender ou comprar um veículo, seja ele novo ou usado. O marketplace permitiria que os usuários fizessem login, anunciassem seus veículos e comprassem.

2. TEMA

Website de busca, anúncios e vendas de veículos de médio e pequeno porte.

3. MOTIVAÇÕES

A ideia partiu de um desejo em comum, nós sempre admiramos e tivemos uma paixão imensa por carros, e com a possibilidade de juntar o útil ao agradável e criar uma plataforma de anúncios e vendas de veículos é como realizar um sonho. E por experiências parecidas ao explorar websites de comércio de veículos foi ficando evidente características e detalhes que nós dois não gostamos e sabíamos que poderíamos fazer melhor com nosso trabalho, diversas mudanças sutis fariam grandes diferenças para a experiência do usuário em uma plataforma digital destas. Portanto a Autochase seria a solução para as necessidades e o aprimoramento do conceito de Marketplace no comércio de veículos na internet.

4. OBJETIVOS

4.1. OBJETIVO GERAL

Desenvolver um website marketplace com foco total em veículos de leve e médio porte.

4.2. OBJETIVOS ESPECÍFICOS

- Facilitar a compra e a venda de veículos de médio e pequeno porte;
- Proporcionar uma plataforma segura e intuitiva para o usuário, buscando alcançar um design sólido, simples e completo, com mecanismos de busca complexos com listas extensas de filtros para pesquisas com marcas e modelos;

5. ESPECIFICAÇÃO DE REQUISITOS

5.1. MÉTODOS DE ESPECIFICAÇÃO DE REQUISITOS

Todas informações serão declaradas pelo usuário no website, desde o cadastro até a criação do anúncio todas informações serão disponibilizadas unicamente por ele de acordo com as instruções deixadas nas páginas.

5.2. REQUISITOS FUNCIONAIS

REF	CASO DE USO	DESCRIÇÃO
RF1	Gerenciar Usuário	Deve permitir, além da criação, a exclusão ou atualização das contas. Também deve permitir que o Administrador do sistema gerencie todos usuários.

RF2	Realizar Login	O sistema deve permitir que usuário realize o login, utilizando seu E-mail e Senha, caso ele não possua uma conta, será exigido a criação de uma.
RF3	Realizar Logout	O sistema deve permitir que usuário realize o logout do sistema, sendo que ele não poderá mais anunciar.
RF4	Gerenciar Anúncio	O sistema deve permitir que o usuário possa criar e gerenciar seu anúncio, sendo possível deletá-lo ou alterá lo, onde ele poderá ou adicionar mais Fotos ao seu anúncio, poderá alterar sua Marca, Modelo, Ano, Chassi, Cor, Combustível, Status, Quilometragem, Versão, Descrição e Preço.
RF5	Consultar Anúncio	O sistema deve permitir que o usuário possa filtrar a pesquisa de um veículo através de palavras chave, modelo, marca, ano etc.
RF6	Gerenciar Marca	Deve permitir que as marcas sejam criadas, editadas ou excluídas pelo administrador do sistema.
RF7	Gerenciar Modelo e Fipe	Deve permitir que os modelos e valores de fipe sejam criados, editados ou excluídos pelo administrador do sistema.
RF8	Gerenciar Cor	Deve permitir que as cores sejam criadas, editadas ou excluídas pelo administrador do sistema.

RF9	Gerenciar Chassi	Deve permitir que os chassis sejam criados, editados ou excluídos pelo administrador do sistema.
RF10	Gerenciar Combustível	Deve permitir que os combustíveis sejam editados ou excluídos pelo administrador do sistema.

5.3. REQUISITOS NÃO FUNCIONAIS

REF	CASO DE USO	DESCRIÇÃO
RNF1	Segurança	O sistema deve criptografar a senha do usuário após o cadastro e manter a privacidade de dados.
RNF2	Segurança	O sistema deve exigir uma senha forte para os usuários , com no mínimo 8 caracteres.
RNF3	Desenvolvimento	O sistema deve ter conexão com o banco de dados MySQL.
RNF4	Desenvolvimento	O sistema utilizará a linguagem de programação HTML, CSS & Javascript.
RNF5	Portabilidade	O sistema deverá rodar somente na plataforma desktop via website.
RNF6	Conexão	O sistema deve ser compatível em dispositivos com acesso a internet.
RNF7	Disponibilidade	O sistema deve estar disponível 99% do tempo.

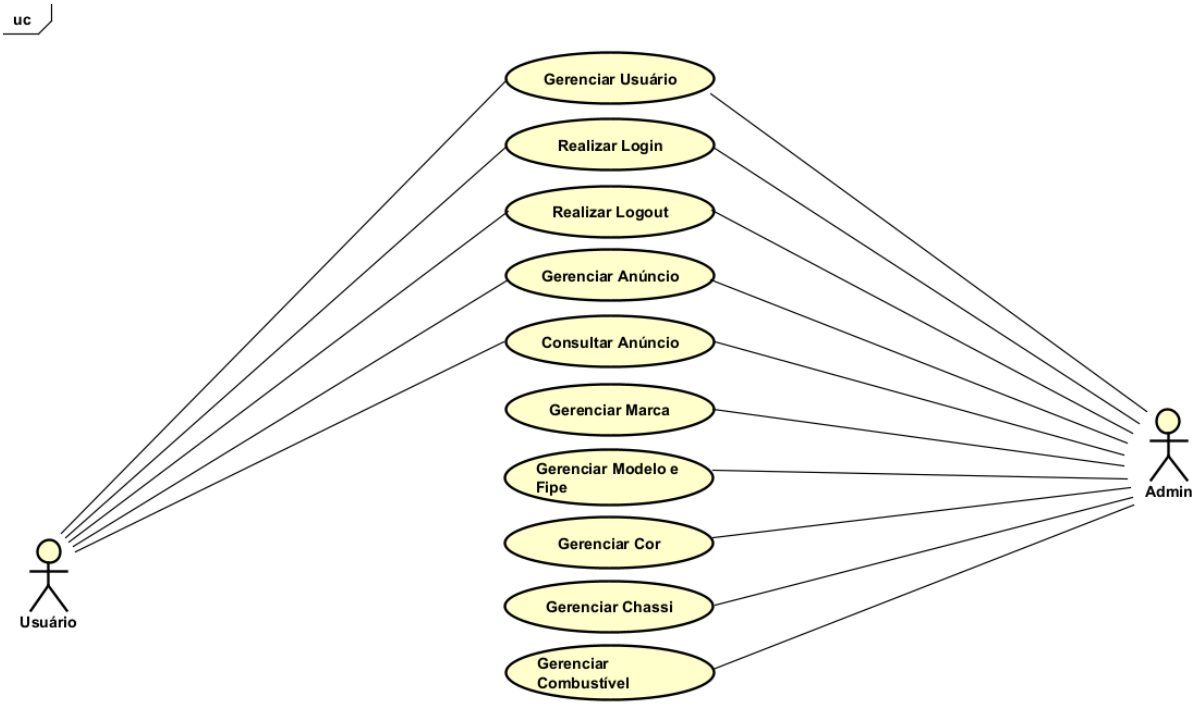
RNF8	Legais	O sistema deverá atender às normas legais, tais como padrões, leis, de cada país onde estiver disponível.

6. MODELAGEM

6.1. MODELO DE CASOS DE USO

A modelagem dos requisitos funcionais do sistema AutoChase é representada pelo diagrama de casos de uso, que ilustra a segregação de responsabilidades entre os atores 'Usuário' e 'Administrador', detalhando no formato expandido essencial às operações de maior complexidade e risco para o negócio. Destaca-se o caso de uso 'Cadastrar Anúncio' (UC05), que implementa uma lógica transacional crítica ao exigir a persistência atômica e simultânea dos registros de veículo e anúncio, assegurando a integridade referencial da base de dados. Complementarmente, o diagrama evidencia a aplicação de regras de segurança baseadas em função (RBAC), onde a manipulação de tabelas de domínio sensíveis, descrita no caso de uso 'Gerenciar Modelo e Tabela FIPE' (UC11), é restrita exclusivamente ao perfil administrativo para garantir a padronização das informações, enquanto operações de consulta e filtragem (UC03) são disponibilizadas aos usuários para viabilizar a dinâmica de compra e venda na plataforma.

Figura 1: Modelo de caso de uso.

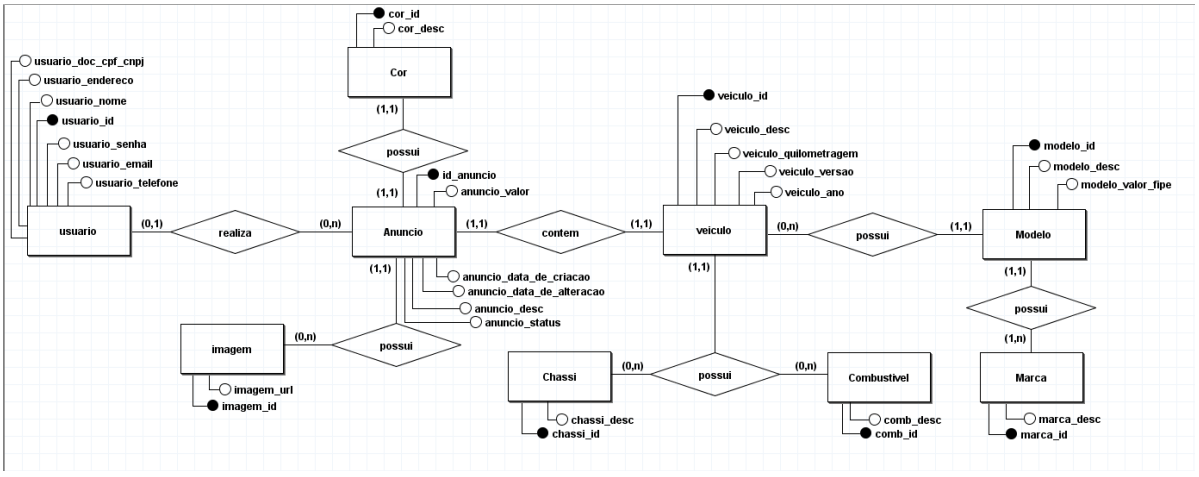


Fonte: Autoria própria.

6.2. MODELAGEM CONCEITUAL DO BANCO DE DADOS.

O diagrama apresentado ilustra o Modelo Conceitual do banco de dados, elaborado para representar as regras de negócio e as relações entre as entidades do sistema de vendas de veículos. O modelo foi desenvolvido visando a integridade das informações e a normalização dos dados.

Figura 2: Diagrama Conceitual do banco de dados.



Fonte: Autoria própria.

O banco de dados do sistema AutoChase foi projetado para dar suporte a um marketplace de anúncios e vendas de veículos de pequeno e médio porte. Ele segue o modelo relacional, com foco em normalização, integridade referencial e facilidade de manutenção, separando dados estáticos (domínio) de dados transacionais (usuários, veículos e anúncios).

A estrutura permite:

- Cadastro e autenticação de usuários
- Criação e gerenciamento de veículos
- Publicação e controle de anúncios
- Padronização de marcas, modelos, cores, combustíveis e tipos de chassi

6.2.1. Usuário:

Responsável por armazenar os dados dos usuários do sistema, tanto compradores quanto vendedores e administradores.

Principais atributos:

- usuario_id (PK): Identificador único do usuário
- usuario_nome: Nome completo
- usuario_email: E-mail de login
- usuario_senha: Senha criptografada
- usuario_nivel_acesso: Define o papel do usuário (ADMIN ou USUARIO)
- usuario_telefone: Telefone para contato
- usuario_endereco: Endereço
- usuario_doc_cnpj: CPF ou CNPJ

Relacionamentos:

- Um usuário pode possuir vários anúncios (1:N com anuncio)

6.2.2. Anúncio:

Representa os anúncios publicados na plataforma, ligando usuários aos veículos.

Principais atributos:

- anuncio_id (PK)
- anuncio_valor: Valor de venda do veículo
- anuncio_data_de_criacao
- anuncio_data_de_alteracao
- anuncio_desc: Descrição detalhada
- anuncio_status: Status do anúncio (ATIVO, INATIVO, VENDIDO)

Chaves estrangeiras:

- fk_usuario_id → usuario
- fk_veiculo_id → veículo

Relacionamentos:

- Cada anúncio pertence a um usuário
- Cada anúncio está vinculado a um veículo

6.2.3. Veículo:

Armazena as informações técnicas e físicas dos veículos cadastrados.

Principais atributos:

- veiculo_id (PK)
- veiculo_quilometragem
- veiculo_ano
- veiculo_versao

Chaves estrangeiras:

- fk_modelo_id → modelo
- fk_cor_id → cor
- fk_chassi_id → chassi
- fk_combustivel_id → combustivel

Relacionamentos:

- Um veículo pode estar associado a um único anúncio
- Cada veículo referência uma marca, modelo, cor, chassi e combustível

6.2.4. Modelo:

Define os modelos de veículos disponíveis no sistema, associados a uma marca.

Principais atributos:

- modelo_id (PK)
- modelo_desc: Nome do modelo
- modelo_valor_fipe: Valor de referência FIPE

Chave Estrangeira:

- fk_marca_id → marca

Relacionamentos:

- Uma marca pode possuir vários modelos (1:N)

6.2.5. Cor:

Tabela auxiliar responsável por padronizar as cores dos veículos.

Principais atributos:

- cor_id (PK)
- cor_desc: Descrição da cor

6.2.6. Marca:

Tabela de domínio que armazena as fabricantes de veículos.

Principais atributos:

- marca_id (PK)
- marca_desc: Nome da marca

Relacionamentos:

- Uma marca pode estar associada a vários modelos

6.2.7. Combustível:

Define os tipos de combustível aceitos no sistema.

Principais atributos:

- combustivel_id (PK)
- comb_desc: Tipo de combustível (Gasolina, Diesel, Flex, etc.)

6.2.8. Chassi:

Representa o tipo de carroceria do veículo.

Principais atributos:

- chassi_id (PK)
- chassi_desc: Tipo de chassi (Sedan, Hatch, SUV, etc.)

6.2.9. Imagem:

Responsável pelo armazenamento das imagens associadas aos anúncios/veículos.

Principais atributos:

- `imagem_id` (PK)
- `imagem_url`: Caminho da imagem no servidor
- `imagem_data_insercao`: Data do upload

Relacionamentos:

- Uma imagem pode estar associada a um anúncio ou veículo, permitindo múltiplas imagens por anúncio

6.3. MODELAGEM LÓGICA DO BANCO DE DADOS.

O diagrama apresentado ilustra o Modelo Lógico do banco de dados Autochase. Nesta etapa, as entidades conceituais foram convertidas em tabelas relacionais, com a definição precisa de tipos de dados (data types), chaves primárias (PK), chaves estrangeiras (FK) e restrições de integridade.

A estrutura foi projetada para garantir a normalização dos dados, evitando redundâncias e assegurando a consistência das informações de vendas de veículos.

Figura 3: Modelagem lógica do banco de dados.

Fonte: Autoria própria.

6.3.1. Atributos de Usuário:

Armazena as informações cadastrais e de autenticação dos usuários do sistema.

- usuario_id (INT, PK)
  Identificador único do usuário no sistema.
- usuario_nome (VARCHAR(30))
  Nome completo do usuário, utilizado para identificação e exibição.
  usuario_senha (VARCHAR(255))
  Senha do usuário armazenada de forma criptografada, garantindo segurança no processo de autenticação.
- usuario_nivel_acesso (ENUM('ADMIN','USUARIO'))
  Define o nível de permissão do usuário, controlando o acesso às funcionalidades administrativas do sistema.

- usuario_email (VARCHAR(50))
  Endereço de e-mail utilizado como login no sistema.
- usuario_telefone (VARCHAR(50))
  Telefone para contato entre comprador e vendedor.
- usuario_endereco (VARCHAR(50))
  Endereço informado pelo usuário no cadastro.
- usuario_doc_cnpj (VARCHAR(25))
  Documento de identificação do usuário, podendo ser CPF ou CNPJ.

6.3.2. Atributos de anúncio:

Responsável por armazenar os dados dos anúncios publicados na plataforma.

- anuncio_id (INT, PK)
  Identificador único do anúncio.
- anuncio_valor (DECIMAL(10,2))
  Valor monetário do veículo anunciado.
  anuncio_data_de_alteracao (TIMESTAMP)
  Data da última alteração realizada no anúncio.
- anuncio_data_de_criacao (TIMESTAMP)
  Data e hora em que o anúncio foi criado.
- anuncio_desc (VARCHAR(50))
  Descrição resumida do anúncio, contendo informações relevantes sobre o veículo.
- anuncio_status (ENUM('ATIVO','INATIVO','VENDIDO'))
  Indica o estado atual do anúncio no sistema.
- fk_usuario_id (INT, FK)
  Chave estrangeira que referencia o usuário responsável pelo anúncio.
- fk_veiculo_id (INT, FK)
  Chave estrangeira que referencia o veículo associado ao anúncio.

6.3.3. Atributos veículo:

Armazena os dados técnicos e físicos dos veículos cadastrados.

- veiculo_id (INT, PK)
  Identificador único do veículo.
- veiculo_quilometragem (DECIMAL(10,2))
  Quilometragem atual do veículo.
- fk_chassi_id (INT, FK)
  Referência ao tipo de chassi/carroceria do veículo.
- fk_cor_id (INT, FK)
  Referência à cor do veículo.
- fk_modelo_id (INT, FK)
  Referência ao modelo do veículo.
- fk_combustivel_id (INT, FK)
  Referência ao tipo de combustível utilizado pelo veículo.
- veiculo_versao (VARCHAR(50))
  Versão ou configuração específica do veículo.
- veiculo_ano (INT)
  Ano de fabricação ou modelo do veículo.

6.3.4. Atributos de Marca:

Tabela de domínio que armazena as marcas de veículos.

- marca_id (INT, PK)
  Identificador único da marca.
- marca_desc (VARCHAR(50))
  Nome da marca do veículo (ex.: Toyota, Ford, Volkswagen).

6.3.5. Atributos de Modelo:

Armazena os modelos de veículos associados às marcas.

- modelo_id (INT, PK)
  Identificador único do modelo.
- modelo_desc (VARCHAR(50))
  Nome do modelo do veículo.

- modelo_valor_fipe (DECIMAL(10,2))
  Valor de referência do modelo segundo a tabela FIPE.
- fk_marca_id (INT, FK)
  Chave estrangeira que associa o modelo a uma marca.

6.3.6. Atributos cor:

Tabela auxiliar responsável pela padronização das cores dos veículos.

- cor_id (INT, PK)
  Identificador único da cor.
- cor_desc (VARCHAR(50))
  Descrição da cor do veículo.

6.3.7. Atributos combustível:

Define os tipos de combustível aceitos no sistema.

- comb_id (INT, PK)
  Identificador único do tipo de combustível.
- comb_desc (VARCHAR(50))
  Descrição do combustível (Gasolina, Diesel, Flex, Elétrico, etc.).

6.3.8. Atributos chassi:

Responsável por armazenar os tipos de carroceria/chassi.

- chassi_id (INT, PK)
  Identificador único do tipo de chassi.
- chassi_desc (VARCHAR(50))
  Descrição do tipo de chassi (Sedan, Hatch, SUV, Coupé, etc.).

6.3.9. Atributos imagem:

Armazena as imagens associadas aos anúncios.

- imagem_id (INT, PK)
  Identificador único da imagem.
  imagem_url (VARCHAR(255))
  Caminho ou URL da imagem armazenada no servidor.
- fk_anuncio_id (INT, FK)
  Chave estrangeira que associa a imagem a um anúncio específico.
  imagem_data_insercao (TIMESTAMP)
  Data e hora do upload da imagem.

7. TECNOLOGIAS UTILIZADAS

O projeto foi construído sobre uma base sólida composta por **PHP** Orientado a Objetos no back-end, **MySQL** para o banco de dados, e um front-end totalmente customizado com **HTML5**, **CSS3** e **JavaScript**, sem dependência de bibliotecas externas.

A linguagem principal do lado do servidor é o **PHP**. No projeto, ele foi utilizado seguindo o paradigma de Orientação a Objetos (POO).

- O PHP é o "motor" da aplicação. Ele gerencia a autenticação dos usuários, processa os formulários de cadastro de veículos, calcula regras de negócio e orquestra a comunicação entre a interface e o banco de dados.

- A escolha pela Orientação a Objetos (evidenciada pela estrutura de classes do projeto) permite um código mais organizado, seguro e fácil de manter.

Para o armazenamento e gestão das informações, foi utilizado o Sistema Gerenciador de Banco de Dados (SGBD) Relacional **MySQL**.

- É responsável pela persistência segura dos dados. Todas as informações críticas — como contas de usuários, detalhes técnicos dos veículos (marca, modelo, chassi) e o histórico dos anúncios — ficam armazenadas em tabelas relacionais, garantindo a integridade das informações mesmo com grande volume de dados.

A interface visual foi construída de forma nativa, sem a utilização de frameworks visuais prontos (como Bootstrap), demonstrando domínio das tecnologias base da web.

- O **HTML5** fornece a estrutura semântica de todas as páginas, enquanto o **CSS3** é responsável pela identidade visual, layout responsivo e customização estética (cores, fontes e espaçamentos) garantindo uma experiência de uso agradável tanto em computadores quanto em dispositivos móveis.

Para dinamizar a navegação e melhorar a experiência do usuário (UX), foi utilizado o **JavaScript** (Vanilla).

- Ele atua no lado do cliente (navegador), permitindo validação de formulário em tempo real e manipulação de elementos da tela sem a necessidade de recarregar a página a todo momento.

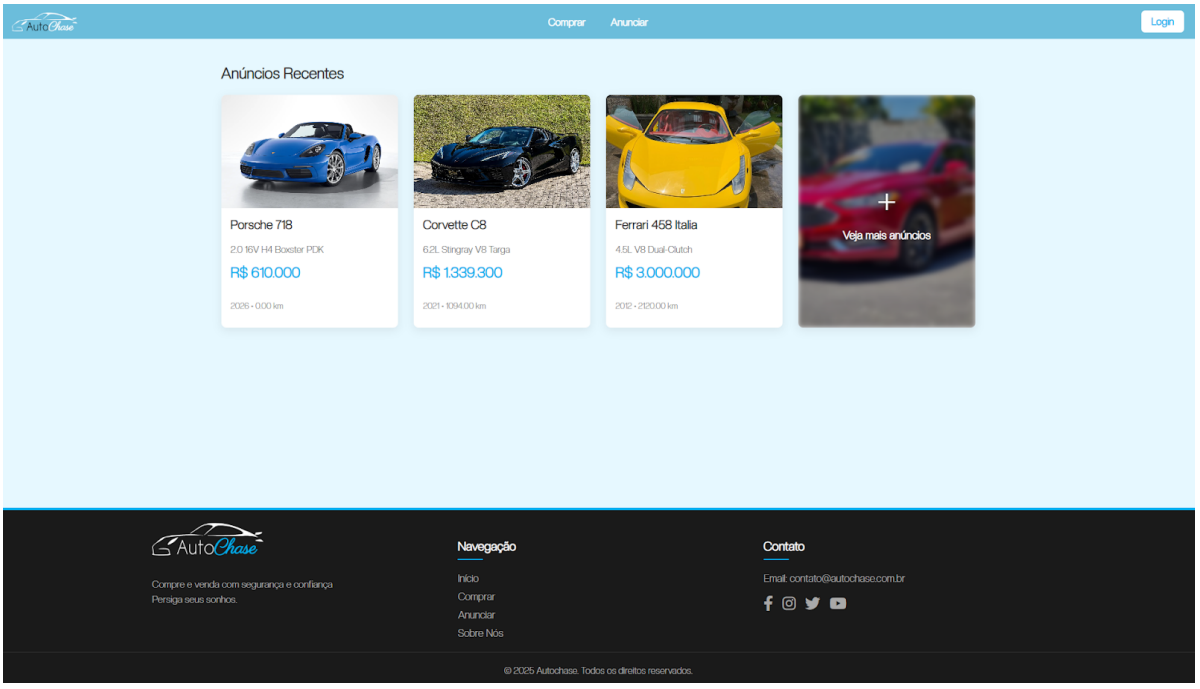
8. DESCRIÇÃO DO SISTEMA

8.1. Tela Inicial

A tela inicial (Figura 4) corresponde à rota raiz do sistema (/), servindo como uma apresentação do sistema Autochase. Apresentando visualmente uma estética clean e moderna, caracterizada pelo uso predominante de tons de azul-ciano e branco no corpo da página, evocando clareza e confiança. Na parte superior, destaca-se um cabeçalho horizontal contendo o logotipo do projeto à esquerda, opções de navegação centralizadas "Comprar", "Anunciar" e um botão de "Login" posicionado à direita, estabelecendo hierarquia visual clara para as ações do usuário.

O conteúdo central é dedicado à seção "Anúncios Recentes", apresentando uma disposição em grid de cartões (cards) informativos. Cada card exibe a imagem do veículo, modelo, especificações técnicas, preço em destaque e quilometragem, facilitando a comparação rápida. O rodapé, em contraste com o restante da interface, utiliza um fundo escuro para delimitar o fim da página, organizando links de navegação, informações de contato e direitos autorais de forma estruturada e acessível.

Figura 4: Tela inicial do sistema Autochase.



Fonte: Autoria própria.

A região superior da interface é delimitada pelo Cabeçalho (Header), que atua como o elemento de navegação persistente da aplicação. À esquerda, encontra-se a identidade visual do sistema, composta pelo logotipo “AutoChase” acompanhado de um ícone estilizado de silhueta automotiva, servindo tanto como reforço do símbolo do projeto quanto como âncora para o retorno à página inicial (Figura 4). Ao centro, a barra de navegação apresenta links textuais diretos para as funcionalidades primárias (“Comprar” e “Anunciar”), utilizando uma tipografia sem serifa branca que garante legibilidade e contraste.

Na extremidade direita, destaca-se o botão de “Login” (Figura 5). Diferentemente dos links centrais, este elemento utiliza um container branco com texto azul (inversão de cores), conferindo-lhe maior peso visual. Essa escolha de design prioriza a ação de autenticação do usuário, destacando-a como uma funcionalidade chave sem poluir a estética minimalista do topo

No corpo da página, a interface estrutura-se sob o título de seção “Anúncios Recentes”, organizando o conteúdo principal em uma disposição horizontal de cartões (cards) sobre um fundo de tom azul suave. Os três primeiros cards atuam como vitrines individuais exibindo anúncios em ordem decrescente, onde o mais recente é localizado sempre à esquerda, contando com veículos de alto desempenho (Porsche 718, Corvette C8 e Ferrari 458) com uma hierarquia visual definida: uma imagem de destaque no topo, seguida pela nomenclatura do modelo e detalhes técnicos de motorização. Logo abaixo, o preço recebe ênfase visual em cor azul para facilitar a leitura rápida, finalizando com metadados de ano e quilometragem no rodapé do componente, permitindo uma comparação direta entre as ofertas.

Ao final dessa sequência, encontra-se o quarto elemento visual, que difere funcionalmente dos anteriores. Este card apresenta uma imagem de fundo com aplicação de filtro de desfoque (blur) e uma máscara escurecida, servindo de base para um ícone de adição (+) centralizado e o texto “Veja mais anúncios”. Esse componente funciona como um ponto de navegação direta ou Call to Action (CTA), indicando intuitivamente ao usuário que a listagem apresentada é apenas uma

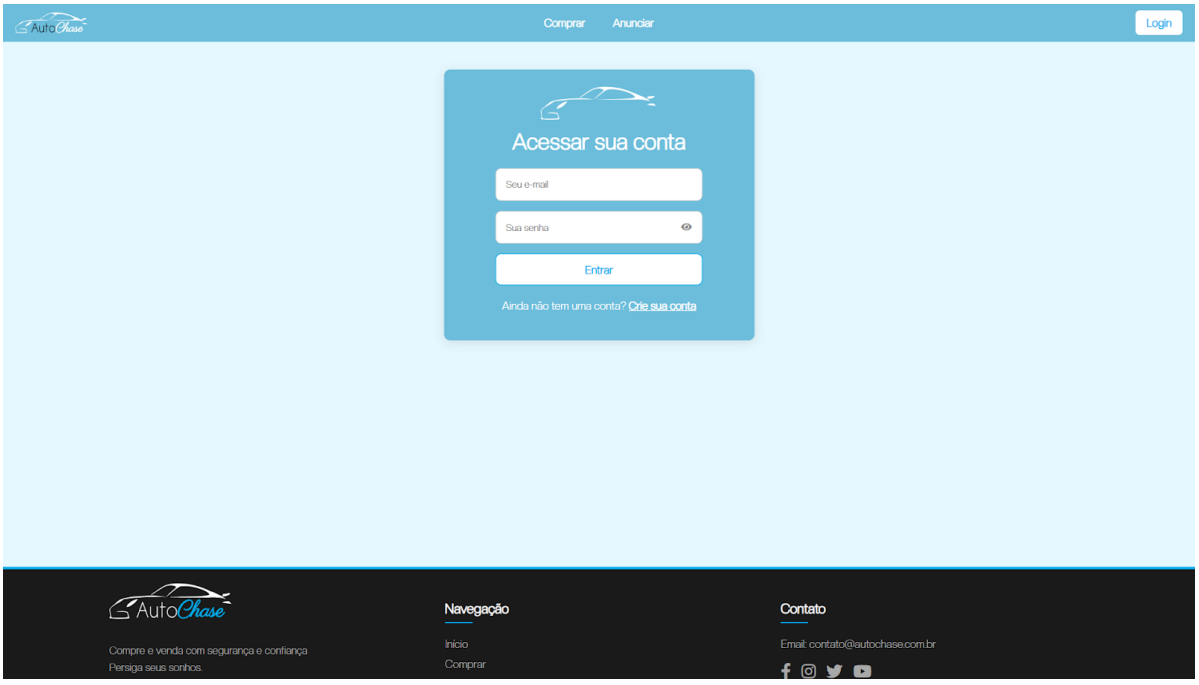
amostra e convidando-o a interagir para acessar o catálogo completo ou a página de busca detalhada do sistema.

8.2. Tela de Login

A interface de autenticação (Figura 5) adota uma arquitetura centrada no usuário, isolando o fluxo de acesso em um componente de cartão (*card*) flutuante, posicionado estrategicamente no centro da *viewport*. Visualmente, este *card* utiliza um preenchimento em azul ciano saturado que contrasta com o fundo mais claro da página, criando uma hierarquia de profundidade e foco.

Dentro deste contêiner, a disposição dos elementos segue uma verticalidade lógica: o logotipo da "AutoChase" no topo reforça a identidade, seguido pelo título "Acessar sua conta". O núcleo da interação consiste em dois campos de entrada (*inputs*) retangulares com fundo branco — um para o e-mail e outro para a senha — garantindo alto contraste e legibilidade imediata sobre o fundo colorido. No campo de senha, nota-se um elemento de interface discreto: um ícone de "olho" alinhado à direita, sugerindo uma funcionalidade de controle de visibilidade. O fluxo é concluído por um botão de ação "Entrar", que inverte as cores (fundo branco, texto azul), e um link de texto sublinhado no rodapé do cartão para criação de novas contas.

Figura 5: Tela de Login do sistema Autochase.



Fonte: Autoria própria.

Expandindo a descrição visual, os campos de e-mail e senha identificados não são elementos soltos, mas componentes integrantes de uma estrutura maior denominada Formulário de Autenticação. Este formulário atua como um contêiner lógico responsável por agrupar e preparar os dados inseridos pelo usuário para o envio. O campo de senha, especificamente, opera por padrão com mascaramento de caracteres (ocultando o texto real), uma medida de segurança essencial para proteger a privacidade da credencial contra visualização por terceiros no ambiente físico.

O ícone de visualização ("olho") descrito anteriormente representa um recurso técnico de Interface de Usuário (UI) voltado à prevenção de erros, permitindo a conferência do dado antes do envio. Já o botão "Entrar" atua como o gatilho de submissão: sua função técnica é coletar as informações do formulário e iniciar o processo de requisição ao servidor. É neste momento que o sistema deixa de ser apenas uma interface visual e interage com o Banco de Dados para validar a existência do usuário, permitindo ou negando o acesso à plataforma. Logo abaixo do botão "Entrar", há uma seção breve de texto em uma linha, sugerindo a criação de

uma conta, caso o usuário não possua nenhuma. Ao ser acessada, essa seção o redireciona para a Tela de Cadastro (Figura 6).

8.3. Tela de Cadastro

A tela de "Crie sua conta" (Figura 6) mantém a coerência estética do sistema, apresentando-se em um cartão centralizado com fundo azul ciano sobre um plano de fundo mais claro. Diferentemente da tela de login, este componente adota um layout mais robusto para acomodar o maior volume de informações. A estrutura interna dos campos abandona a verticalidade simples e adota uma disposição em grade (*grid*) de duas colunas, otimizando o espaço horizontal.

Os seis campos de entrada estão organizados em pares lógicos: na primeira linha, identificação pessoal e contato digital ("Nome completo" e "Seu melhor e-mail"); na segunda, contato físico ("Telefone" e "Endereço"); e na terceira, segurança e identificação fiscal ("Crie uma senha" com o ícone de visibilidade e "CPF ou CNPJ"). O botão de ação principal, "Criar Conta", ocupa toda a largura inferior do formulário, utilizando a inversão de cores (fundo branco) para destacar a conclusão do processo. Por fim, localizado estrategicamente no rodapé do cartão, encontra-se o link de redirecionamento para o Login (Figura 5). Este elemento oferece uma rota de saída clara para usuários que acessaram a tela por engano, permitindo o retorno imediato à área de autenticação sem a necessidade de utilizar os controles do navegador.

Figura 6: Tela de Cadastro do sistema Autochase.

Fonte: Autoria própria.

Esta interface representa um Formulário de Registro de Entidade enriquecido por validações no lado do cliente (*Client-Side*). A arquitetura de interação depende diretamente de um controlador JavaScript que manipula o DOM (*Document Object Model*) para garantir a integridade dos dados antes do envio ao servidor.

O destaque técnico reside na lógica de Máscara Híbrida Dinâmica aplicada ao campo "CPF ou CNPJ". O algoritmo implementado não apenas formata o texto, mas realiza uma sanitização de entrada (removendo caracteres não numéricos via *Regex*) e decide automaticamente qual máscara aplicar com base no comprimento da *string* (alternando entre os padrões de 11 dígitos para CPF e 14 para CNPJ). Além da formatação, o sistema conta com um mecanismo de Feedback Visual Síncrono: conforme o usuário digita, uma janela de texto surge à direita do campo de inserção e a função verifica a completude do dado, injetando classes de estado (status-success ou status-error) em um componente de *popup* oculto. Isso fornece ao usuário uma resposta imediata sobre a validade do tamanho do documento, sem a necessidade de recarregar a página.

O mesmo padrão de expressões regulares é aplicado ao campo "Telefone" para garantir a padronização, enquanto o campo de senha utiliza uma manipulação simples de atributos de input para alternar a visibilidade dos caracteres. Encerrando

o fluxo lógico da interface, o controle de retorno ao Login (Figura 5) é implementado através de uma âncora de navegação. Tecnicamente, este elemento não dispara a submissão do formulário, mas altera a rota da aplicação (roteamento), limpando o contexto de registro atual e redirecionando a requisição para a *view* de autenticação já existente.

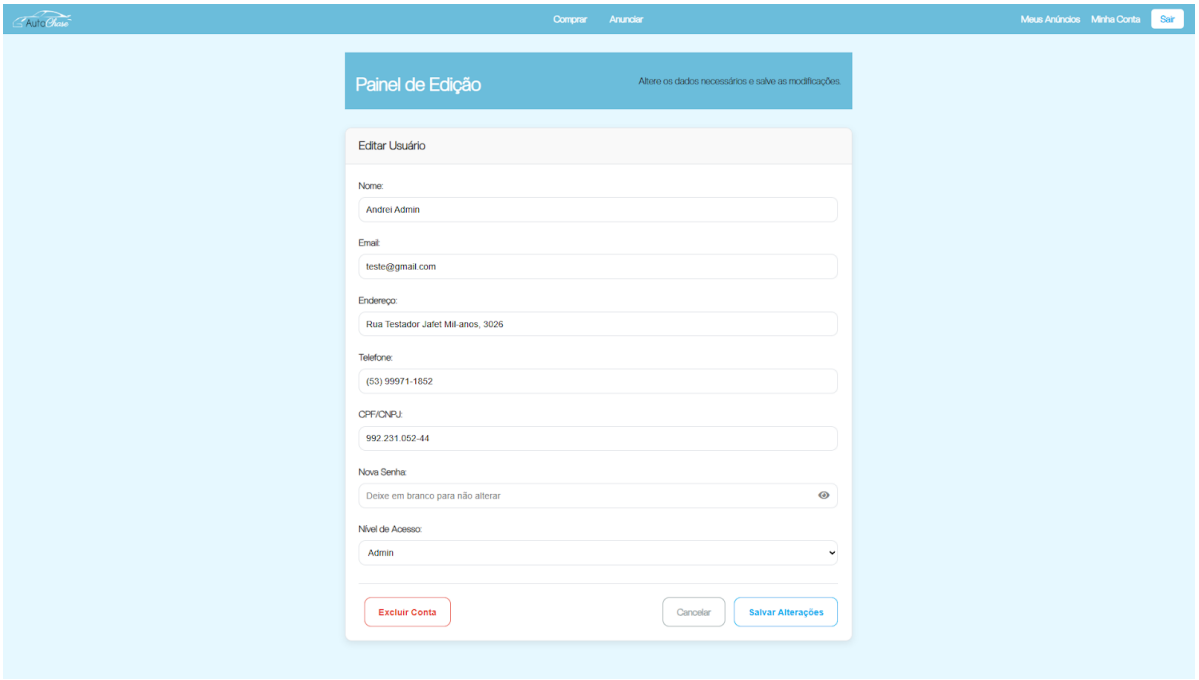
8.4. Tela de Edição do Usuário

A interface apresentada (Painel de Edição) (Figura 7) reflete o estado da aplicação quando o usuário já está autenticado. O cabeçalho superior, mantendo o azul padrão da marca, agora exibe opções de navegação interna ("Meus Anúncios", "Minha Conta") e um botão de "Sair" à direita, substituindo a antiga opção de login. Logo abaixo, uma faixa de título em azul intermediário introduz a seção "Painel de Edição" com uma breve instrução de uso, separando visualmente o cabeçalho do conteúdo principal.

O formulário de edição é centralizado em um *card* branco e caracteriza-se pelo preenchimento prévio (*pre-filled*) dos dados. Campos como "Nome", "Email", "Telefone" e "CPF/CNPJ" já exibem as informações do usuário "Andrei Admin", permitindo a visualização imediata do cadastro atual. Destacam-se dois campos diferenciados: "Nova Senha", que possui um texto de instrução como *placeholder* ("Deixe em branco para não alterar") e o ícone de visibilidade; e "Nível de Acesso", que é apresentado como uma lista de seleção (*dropdown*), exibindo o valor "Admin".

No rodapé do formulário, a hierarquia de ações é definida pelas cores: à esquerda, um botão com contorno vermelho "Excluir Conta" indica uma ação destrutiva e perigosa; à direita, o botão cinza "Cancelar" oferece uma saída neutra, e o botão azul "Salvar Alterações" destaca a ação primária de confirmação.

Figura 7: Tela de Edição da Conta do Usuário do sistema Autochase.



Fonte: Autoria própria.

Aqui a tela materializa duas operações fundamentais do acrônimo CRUD: o Read (Leitura) e o Update (Atualização), além de oferecer o gatilho para o Delete (Exclusão).

Tecnicamente, ao carregar esta página, o *backend* executa uma consulta ao banco de dados (query SELECT) utilizando o ID da sessão do usuário logado para popular os atributos value dos *inputs* HTML. Isso garante que o usuário edite a partir do estado atual dos dados. O campo "Nível de Acesso" sugere a implementação de um sistema de Controle de Acesso Baseado em Funções (RBAC), onde o valor selecionado no elemento <select> definirá as permissões do usuário no sistema. As máscaras de JavaScript (CPF e Telefone) que analisamos anteriormente continuam atuando aqui: elas formatam visualmente os dados vindos do banco para manter a consistência na exibição.

Um ponto crítico de lógica de *backend* reside no campo "Nova Senha". Diferente dos demais, ele exige uma validação condicional no servidor: se o campo for submetido vazio, a senha antiga (hash) é mantida no banco; se preenchido, o sistema deve gerar um novo *hash* criptográfico antes de realizar o UPDATE. Por fim,

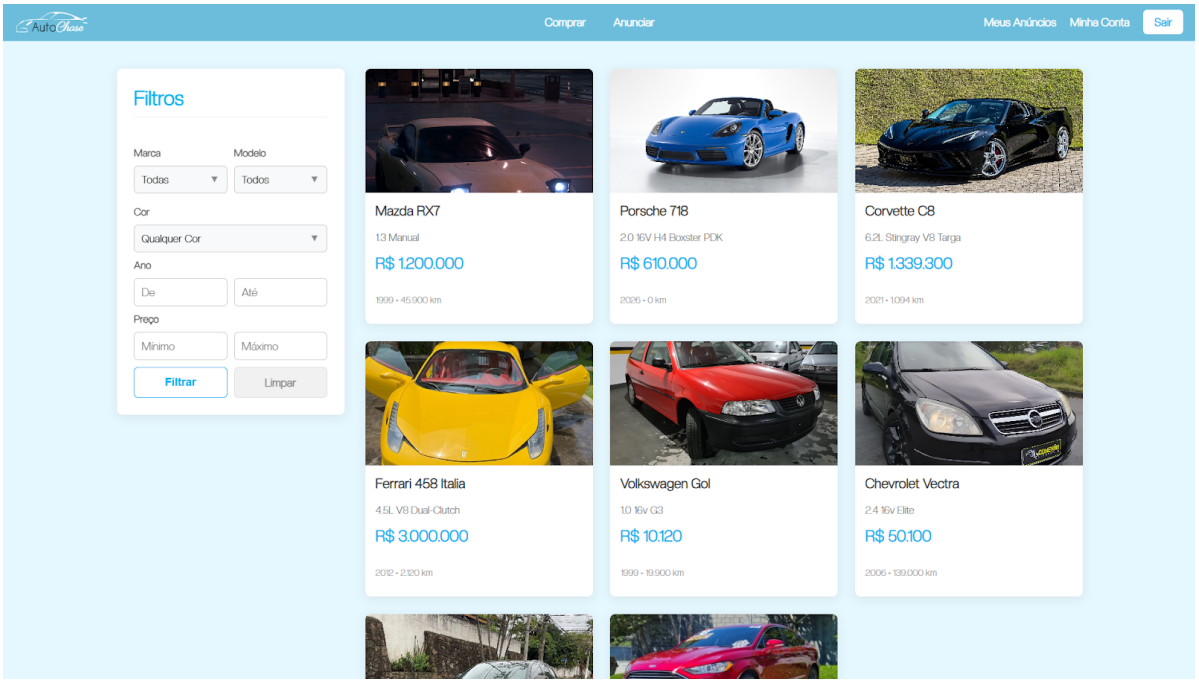
os botões de ação disparam fluxos distintos: "Salvar" submete o formulário via método HTTP POST para persistir as mudanças, enquanto "Excluir Conta" deve acionar uma rota de remoção definitiva, idealmente precedida por um alerta de confirmação em JavaScript (confirm()) para evitar perda acidental de dados, encerrando a sessão através da função de *logout* (botão "Sair" no cabeçalho).

8.5. Tela de Compra

A tela de Catálogo de Veículos (Figura 8) estrutura-se em um *layout* assimétrico de duas colunas, priorizando a exibição dos produtos enquanto mantém as ferramentas de busca acessíveis. À esquerda, destaca-se o painel lateral "Filtros", um cartão branco flutuante que agrupa os critérios de seleção. Este painel organiza os controles de forma hierárquica: seletores (*dropdowns*) para "Marca", "Modelo" e "Cor" no topo, seguidos por campos de intervalo numérico para "Ano" (De/Até) e "Preço" (Mínimo/Máximo). No rodapé do painel, dois botões controlam a ação: "Filtrar" (com borda azul) e "Limpar" (em cinza), permitindo aplicar ou resetar os critérios.

A área principal, à direita, ocupa a maior parte da tela e exibe os resultados em uma grade (*grid*) de cartões. Cada cartão segue o padrão visual estabelecido: imagem do veículo em destaque, nome do modelo, especificações técnicas breves, preço em cor azul ciano para ênfase e, por fim, o ano e a quilometragem discretos na parte inferior. A barra de navegação superior confirma que o usuário está logado, oferecendo acesso rápido a "Meus Anúncios" e "Minha Conta".

Figura 8: Tela de Compra do sistema Autochase.



Fonte: Autoria própria.

O funcionamento desta interface baseia-se na construção dinâmica de consultas ao banco de dados, onde o painel lateral atua como um coletor de parâmetros para a cláusula WHERE do SQL. A interação entre os campos "Marca" e "Modelo" implementa a lógica de dependência de dados observada no código JavaScript: ao selecionar uma marca, um evento de mudança (*change event*) dispara uma função que limpa e repopula o seletor de modelos apenas com as opções correspondentes àquela fabricante, prevenindo combinações inválidas.

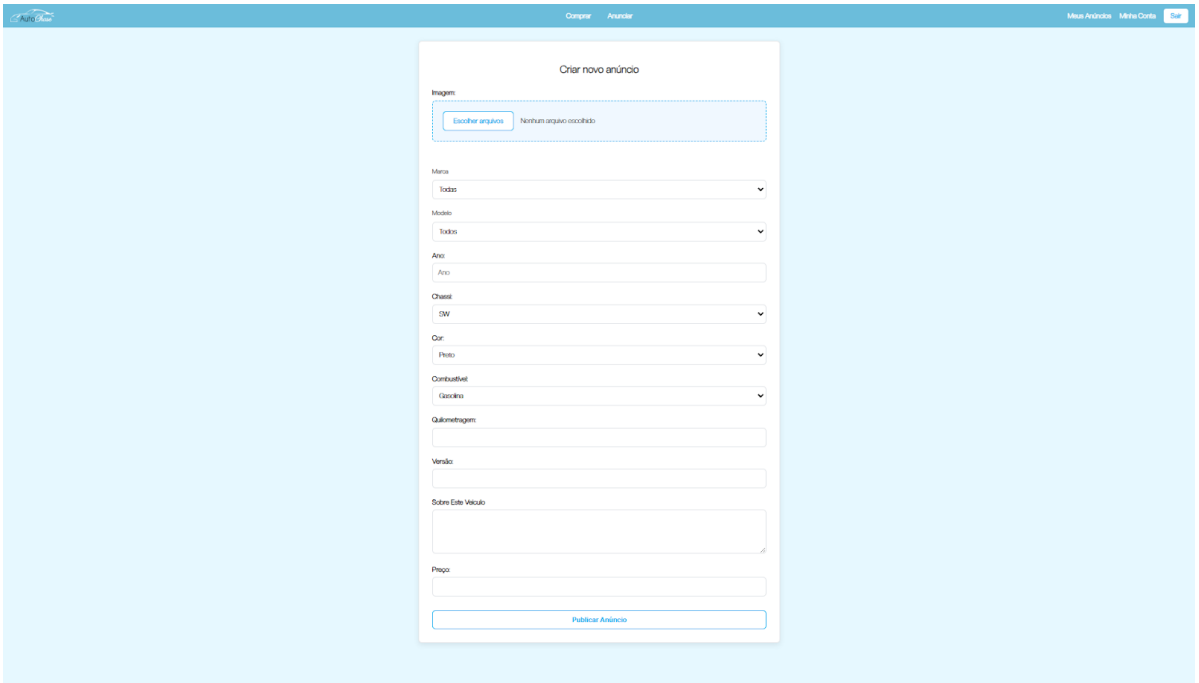
A renderização da grade de veículos é realizada através de uma estrutura de repetição no *backend* (como um laço foreach em PHP), que itera sobre o *array* de resultados retornado pelo banco. Cada iteração injeta os dados do objeto veículo (imagem, preço, especificações) no esqueleto HTML do cartão. Ao clicar no botão "Filtrar", o formulário submete os valores preenchidos, recarregando a página ou atualizando a grade via AJAX com o subconjunto de dados que atende aos critérios de preço, ano e características físicas definidos pelo usuário.

8.6. Tela de Criação do Anúncio

A interface de "Criar novo anúncio" (Figura 9) centraliza o fluxo de entrada de dados em um formulário extenso vertical, contido em um cartão branco sobre o fundo azul padrão do sistema. O topo do formulário prioriza o apelo visual do produto, oferecendo uma área de *upload* de imagens destacada por uma borda azul tracejada e um botão "Escolher arquivos", sugerindo uma funcionalidade de arrastar e soltar (*drag-and-drop*).

Abaixo da imagem, a coleta de dados segue uma estrutura hierárquica mista. Utilizam-se menus suspensos (*dropdowns*) para informações categóricas padronizadas ("Marca", "Modelo", "Chassi", "Cor", "Combustível") e campos de texto livre para dados variáveis ("Ano", "Quilometragem", "Versão", "Preço"). Um campo de texto expandido (*textarea*) denominado "Sobre Este Veículo" permite a inserção de uma descrição detalhada. A interação é finalizada pelo botão "Publicar Anúncio" na parte inferior, que segue o padrão visual de contorno azul (*outline*), indicando a ação de confirmação.

Figura 9: Tela de Anúncio do sistema Autochase.



Fonte: Autoria própria.

A implementação técnica deste formulário difere das telas anteriores devido à necessidade de manipular arquivos binários junto com dados textuais. Para isso, a tag <form> deve, obrigatoriamente, utilizar o atributo enctype="multipart/form-data", permitindo que a imagem selecionada seja enviada via requisição HTTP POST para processamento no servidor.

A lógica de interdependência entre "Marca" e "Modelo" reaparece aqui: o seletor de modelos permanece desabilitado ou vazio até que o evento change no seletor de marcas dispare a função JavaScript carregarModelos, garantindo a integridade relacional dos dados antes mesmo da submissão. Os campos numéricos, especificamente "Preço" e "Quilometragem", integram as funções de máscara (formatarMoeda e sanitização de regex) detalhadas anteriormente, convertendo a entrada do usuário em formatos compatíveis com os tipos de dados do banco (DECIMAL ou INT). Ao acionar o botão "Publicar Anúncio", o *backend* recebe o *payload*, valida os campos obrigatórios e executa uma instrução SQL INSERT para criar o novo registro na tabela de veículos, associando-o ao ID do usuário autenticado na sessão.

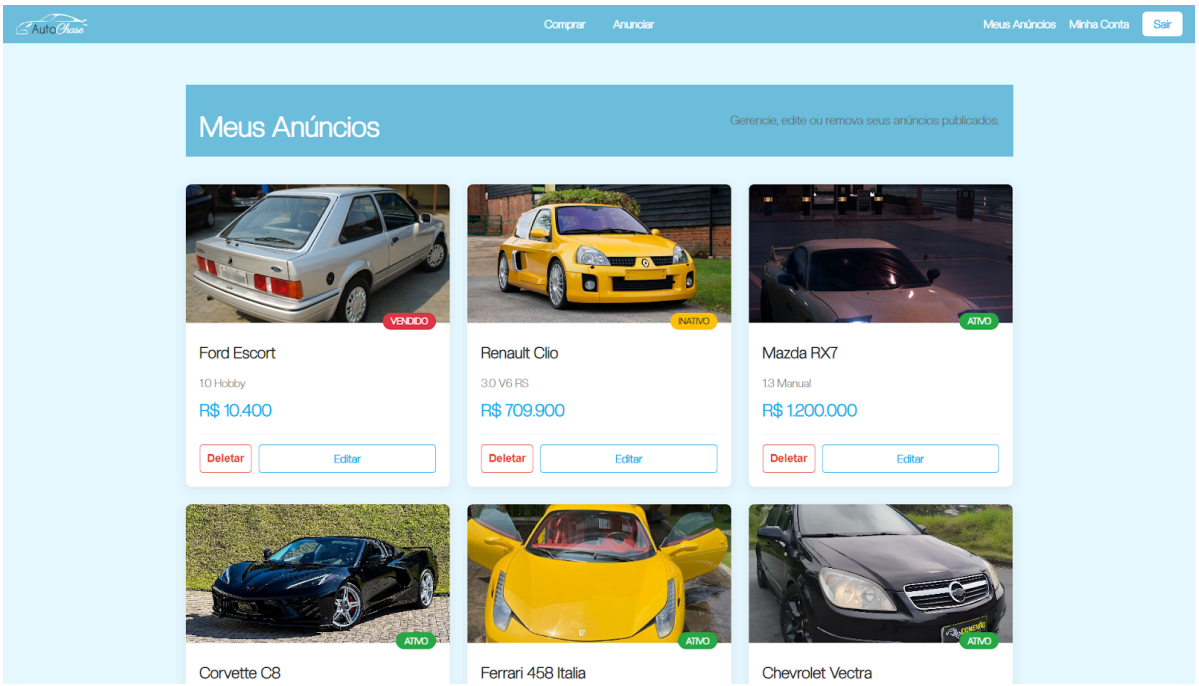
8.7. Tela Meus Anúncios

A página Meus Anúncios (Figura 10) funciona como o painel de controle administrativo do usuário dentro da plataforma. O layout inicia com um cabeçalho de seção distinto em azul, contendo o título e uma breve descrição funcional, separando visualmente esta área de gestão do restante da navegação. O corpo da página organiza os veículos cadastrados pelo usuário em uma grade de cartões idêntica à do catálogo principal, mas com adições cruciais de interface.

Cada cartão apresenta uma etiqueta de status (*badge*) sobreposta no canto inferior direito da imagem do veículo, utilizando um sistema de cores semafórico para indicar a situação do anúncio: vermelho para Vendido, amarelo para Inativo e verde para Ativo. A parte inferior dos cartões substitui as informações de quilometragem por botões de ação direta: Deletar, estilizado com contorno vermelho, e Editar, com contorno azul. Essa disposição permite que o usuário identifique

rapidamente o estado de seu inventário e execute ações de manutenção com um único clique.

Figura 10: Tela Meus Anúncios do sistema Autochase.



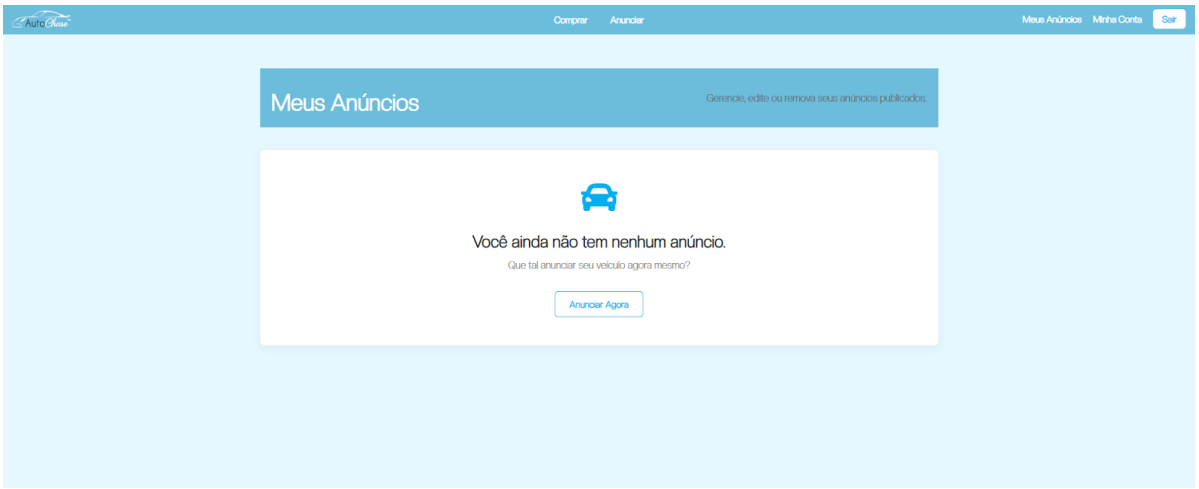
Fonte: Autoria própria.

A construção visual desta tela apoia-se em um Sistema de Grid Fluido. Diferente de tabelas rígidas, a estrutura foi projetada para ser intrinsecamente responsiva: ela calcula automaticamente quantos cartões cabem na linha com base no tamanho da tela do usuário, reorganizando o layout dinamicamente para garantir a melhor visualização em monitores, sem quebras visuais.

Um ponto de sofisticação na Experiência do Usuário (UX) é o gerenciamento de Camadas de Interação. Para facilitar a navegação, toda a área do cartão funciona como um link clicável. No entanto, para evitar conflitos, os botões de "Editar" e "Deletar" foram tecnicamente isolados em uma camada superior (superposta). Isso garante que, ao clicar nessas ações específicas, o sistema priorize o comando de manutenção e bloqueie o redirecionamento padrão do cartão, resolvendo um problema comum de sobreposição de eventos de clique.

Além disso, a interface trata de forma inteligente o Estado Vazio (Empty State). O sistema detecta quando a consulta ao banco de dados não retorna veículos e, automaticamente, altera a estrutura do grid. Em vez de mostrar uma tela em branco ou quebrada, ele renderiza um componente centralizado que ocupa toda a largura disponível, exibindo uma mensagem amigável e um botão de incentivo, guiando o usuário para a ação de criar seu primeiro anúncio (Figura 11), assim redirecionando o usuário para a Tela de Criação do Anúncio (Figura 9).

Figura 11: Tela Meus Anúncios vazia do sistema Autochase.



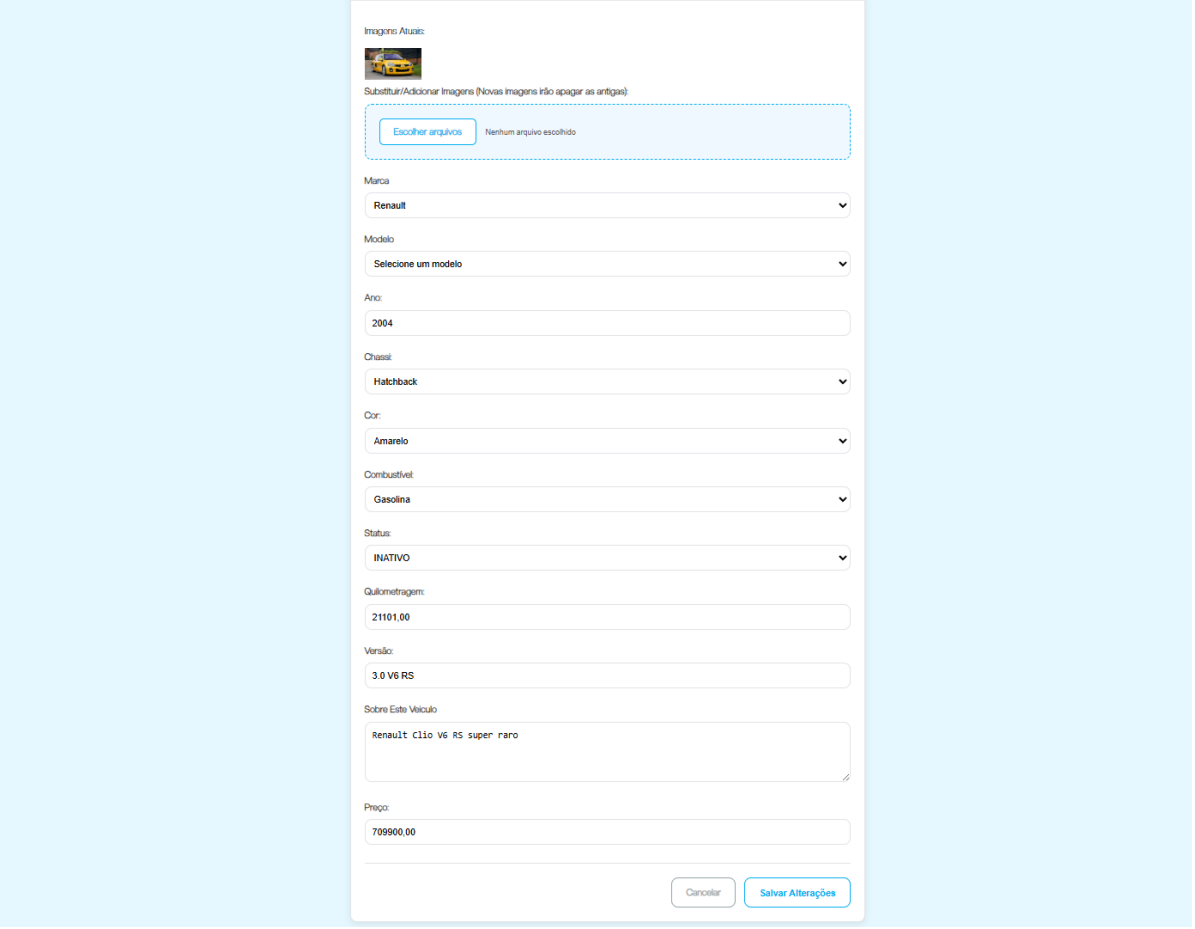
Fonte: Autoria própria.

8.8. Tela Editar Meus Anúncios

A interface de "Edição de Anúncio" (Figura 12) reaproveita a estrutura visual da criação de anúncios (Figura 9), mas adapta-se funcionalmente para a manutenção de registros existentes. A principal distinção visual ocorre no topo do formulário, onde uma seção denominada "Imagens Atuais" exibe as miniaturas das fotos já cadastradas, permitindo que o usuário visualize o ativo antes de qualquer modificação. Logo abaixo, a área de upload contém um aviso textual importante informando que a inserção de novos arquivos substituirá os antigos.

Os campos de dados subsequentes (como "Marca", "Ano", "Preço" e "Descrição") aparecem preenchidos com as informações originais do veículo, facilitando a conferência. Um elemento exclusivo desta tela é o campo seletor de "Status", posicionado centralmente, que exibe o estado atual (neste caso, "INATIVO"), permitindo ao usuário controlar a visibilidade do anúncio sem excluí-lo. O rodapé mantém a consistência com botões de ação para "Cancelar" ou "Salvar Alterações".

Figura 12: Tela Editar Meus Anúncios do sistema Autochase.



Fonte: Autoria própria.

Esta tela materializa a operação de *Update* do ciclo *CRUD*. O carregamento inicial da página depende de uma requisição que executa uma consulta no banco de dados e popula os atributos *value* dos *inputs* HTML, garantindo que o usuário edite a partir do estado atual dos dados.

A lógica de manipulação de arquivos difere da criação: o *backend* precisa implementar uma rotina de verificação que, ao detectar o *upload* de novas imagens durante a edição, remove fisicamente os arquivos antigos do servidor e atualiza as referências no banco de dados, otimizando o armazenamento. O campo "Status" atua como um interruptor lógico (*flag*); sua alteração para "Inativo" faz com que o

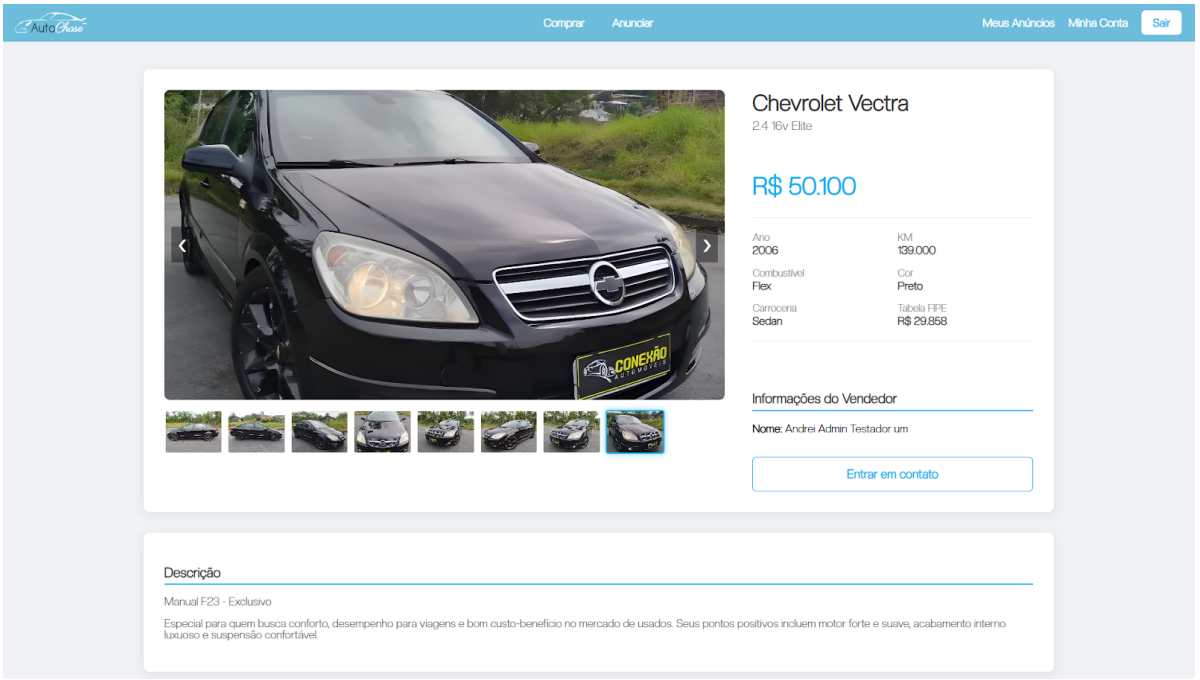
veículo seja ignorado nas consultas *SQL* do catálogo público (que filtram por *status* ativo), mas permaneça no banco de dados para gestão histórica. As máscaras de *JavaScript* para moeda e quilometragem atuam novamente aqui, convertendo o formato de armazenamento do banco (ponto flutuante) para o formato visual (com vírgulas e pontos) no momento do carregamento da página.

8.9. Tela Detalhes do Anúncio

A tela de "Detalhes do Anúncio" (Figura 13) é a página de conversão principal da plataforma, projetada para apresentar a totalidade das informações de um veículo. O *layout* principal é estruturado em um contêiner centralizado que utiliza uma divisão assimétrica: à esquerda, ocupando a maior parte da largura, encontra-se a galeria de imagens; à direita, reside o bloco de informações comerciais.

A galeria é composta por uma imagem principal de grande formato, equipada com setas laterais de navegação (*carousel*), e uma faixa inferior de miniaturas que permite a seleção rápida de fotos. Ao lado, o painel de informações exibe hierarquicamente o título do veículo, a versão e o preço em destaque na cor azul ciano. Logo abaixo, uma grade de especificações técnicas organiza dados como "Ano", "KM", "Combustível" e "Tabela FIPE". A seção superior encerra-se com os dados do vendedor e o botão "Entrar em contato". Diferente das versões anteriores, a descrição textual detalhada do veículo foi movida para um contêiner exclusivo de largura total, posicionado abaixo do bloco principal, garantindo maior espaço para leitura.

Figura 13: Tela do Anúncio e suas informações do sistema Autochase.



Fonte: Autoria própria.

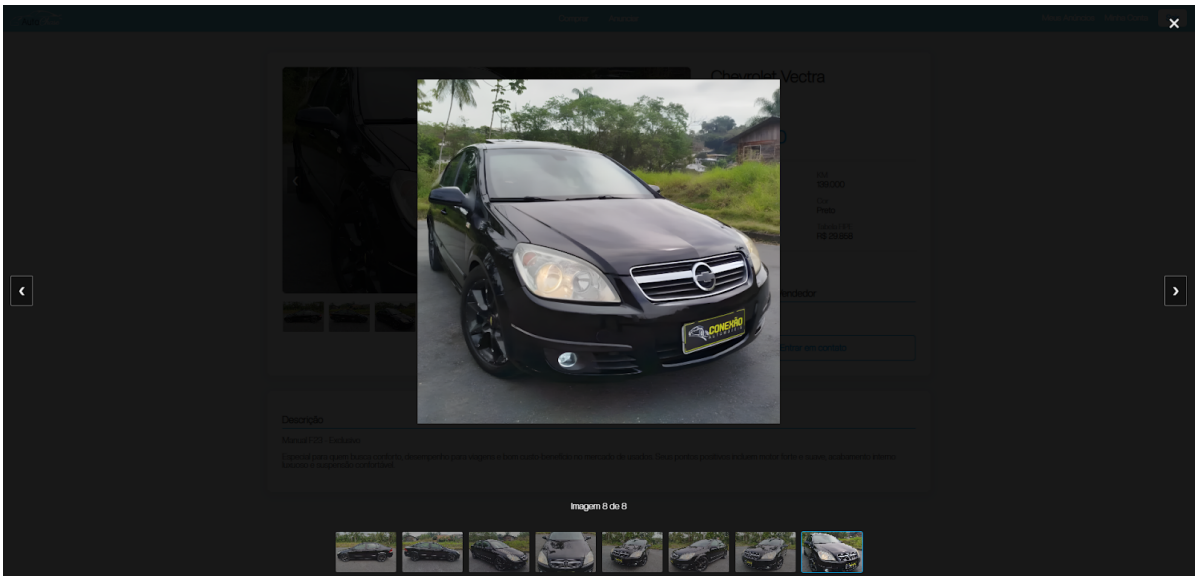
O funcionamento desta página inicia-se no *backend* com a captura do parâmetro *anuncio_id* via método *GET*. O *script PHP* valida a existência deste ID e instancia as classes *Anuncio* e *Foto* para realizar as consultas ao banco de dados. Caso o ID seja inválido ou o anúncio não exista, o sistema implementa uma proteção de redirecionamento automático para a página inicial.

A construção visual utiliza *CSS Grid* para dividir a tela na proporção de 2/3 para imagens e 1/3 para informações, garantindo responsividade. A interatividade da galeria é híbrida: o *PHP* gera um *array* de URLs de imagens dentro de uma tag *script*, injetando os dados diretamente no *frontend*. Isso permite que funções *JavaScript* manipulem o atributo *src* da imagem principal instantaneamente, sem novas requisições ao servidor.

Além disso, o código revela a implementação de um *Lightbox Modal* (uma janela de sobreposição oculta por padrão com *display: none*). Quando o usuário clica em uma imagem, o *JavaScript* altera o estilo para exibir o modal em tela cheia (Figura 14), utilizando um *z-index* elevado para sobrepor todo o conteúdo do site. O botão "Entrar em contato" armazena o número do vendedor em um atributo de

dados (*data-telefone*), separando a informação da lógica de clique.

Figura 14: Tela do Anúncio com Imagens em Tela Cheia do sistema Autochase.



Fonte: Autoria própria.

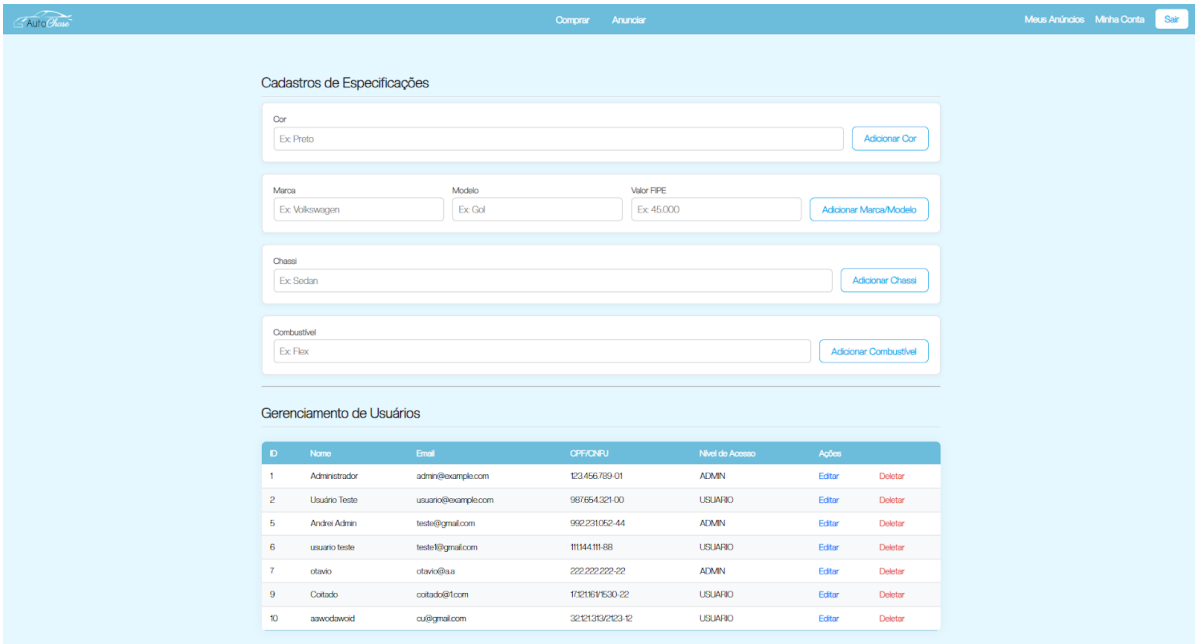
8.10. Tela Inicial do Painel de Administrador

A interface apresentada corresponde ao "Painel Administrativo" (Figura 15), uma área restrita integrada à estrutura principal do sistema, projetada para o gerenciamento global de dados. Visualmente, a tela divide-se em duas macro-seções verticais sobre o fundo azul claro padrão da identidade visual.

A parte superior, intitulada "Cadastros de Especificações", é composta por quatro blocos de formulários independentes e horizontais. Cada bloco é dedicado a uma categoria de dado veicular: "Cor", "Marca" (agrupada com "Modelo" e "Valor FIPE"), "Chassi" e "Combustível". Todos seguem um padrão ergonômico consistente: campos de entrada de texto (*inputs*) à esquerda e um botão de ação correspondente à direita (ex: "Adicionar Cor", "Adicionar Combustível"), facilitando a inserção rápida de novos parâmetros no sistema.

A parte inferior exibe a seção "Gerenciamento de Usuários", estruturada na forma de uma tabela de dados (*data table*). O cabeçalho da tabela, em azul sólido, define as colunas: "ID", "Nome", "Email", "CPF/CNPJ" e "Nível de Acesso". A última coluna, "Ações", contém links textuais para "Editar" e "Deletar", permitindo a moderação direta das contas cadastradas na plataforma.

Figura 15: Tela de Cadastro de Especificações do sistema Autochase.



Fonte: Autoria própria.

Esta seção da tela atua como o *dashboard* de controle mestre. Sua renderização deve ser obrigatoriamente precedida por uma validação de segurança no *backend* (verificação de *Session*), assegurando que apenas usuários com a permissão "ADMIN" tenham acesso a estas ferramentas, implementando o conceito de Controle de Acesso Baseado em Funções (*RBAC*).

Os formulários superiores são responsáveis por alimentar as tabelas de domínio do banco de dados. Estas tabelas armazenam os dados estáticos que povoam os menus suspensos (*dropdowns*) nas telas de cadastro de veículos. O formulário que agrupa "Marca", "Modelo" e "Valor FIPE" sugere uma lógica de inserção relacional: ao submeter este formulário, o sistema cria o registro da marca (se não existir) e, em seguida, insere o modelo vinculando-o a essa marca através de uma chave estrangeira (*Foreign Key*).

A seção de usuários, por sua vez, executa uma operação de leitura em lote (*Read all*). O *script* percorre os registros da tabela de usuários através de um laço de repetição (*loop*), gerando dinamicamente as linhas da tabela *HTML*. Os botões de ação na coluna final utilizam a passagem de parâmetros via *URL* (método *GET* com o ID do usuário) para direcionar o administrador às rotas específicas de edição ou

exclusão lógica/física da conta alvo.

Figura 16: Listagem de Veículos do sistema Autochase.

Listagem de Veículos

ID do Veículo	Quilometragem	Marca	Modelo	Versão	Carrocaria	Cor	Combustível	Ano
26	10.892	Audi	RS6	4.0 Avant V8 Twinturbo MHEV Tiptronic	SW	Preto	Gasolina	2021
28	19.900	Volkswagen	Gol	1.0 16v G3	Hatchback	Vermelho	Gasolina	1999
29	2.120	Ferrari	458 Italia	4.5L V8 Dual-Clutch	Coupé	Amarelo	Gasolina	2012
31	1.094	Corvette	C8	6.2L Stingray V8 Targa	Coupé	Preto	Gasolina	2021
32	0	Porsche	718	2.0 16V H4 Boxster PDK	Spyder	Preto	Gasolina	2026
33	45.900	Mazda	RX7	1.3 Manual	Coupé	Branco	Gasolina	1999
37	21.101	Renault	Clio	3.0 V6 RS	Hatchback	Amarelo	Gasolina	2004
38	259.000	Ford	Escort	1.0 Hobby	Hatchback	Branco	Gasolina	1986
25	123.000	Ford	Fusion	2.0 TITANIUM AWD 16V	Sedan	Vermelho	Híbrido	2018
27	139.000	Chevrolet	Vectra	2.4 16v Elite	Sedan	Preto	Flex	2006
27	139.000	Chevrolet	Vectra	2.4 16v Elite	Sedan	Preto	Flex	2006
27	139.000	Chevrolet	Vectra	2.4 16v Elite	Sedan	Preto	Flex	2006
27	139.000	Chevrolet	Vectra	2.4 16v Elite	Sedan	Preto	Flex	2006
27	139.000	Chevrolet	Vectra	2.4 16v Elite	Sedan	Preto	Flex	2006
27	139.000	Chevrolet	Vectra	2.4 16v Elite	Sedan	Preto	Flex	2006
27	139.000	Chevrolet	Vectra	2.4 16v Elite	Sedan	Preto	Flex	2006
27	139.000	Chevrolet	Vectra	2.4 16v Elite	Sedan	Preto	Flex	2006

Fonte: Autoria própria.

A seção "Listagem de Veículos" (Figura 16) exibe uma tabela administrativa centralizada que compila o inventário global do sistema. O cabeçalho, destacado em azul, organiza as informações técnicas em colunas essenciais como "ID do Veículo", "Marca", "Modelo" e "Ano". As linhas subsequentes apresentam os dados de forma tabular e simples, permitindo uma conferência visual rápida de todos os ativos cadastrados, sem elementos de interação direta nesta visualização.

Esta interface opera como um *Data Grid* de auditoria. O *backend* processa uma consulta *SQL* abrangente (utilizando *JOINs* para converter chaves estrangeiras em nomes legíveis de marcas e cores) e renderiza o resultado em *HTML*. Trata-se de uma funcionalidade *Read-only* (somente leitura), essencial para que o

administrador valide a consistência dos dados e identifique duplicidades ou erros no banco de dados.

Figura 17: Gerenciamento de Marcas do sistema Autochase.

Gerenciamento de Marcas

ID da marca	Nome da marca	Ações	
1	Toyota	Editar	Deletar
2	Honda	Editar	Deletar
3	Ford	Editar	Deletar
4	Volkswagen	Editar	Deletar
9	Audi	Editar	Deletar
10	Mazda	Editar	Deletar
15	Chevrolet	Editar	Deletar
17	Ferrari	Editar	Deletar
18	Corvette	Editar	Deletar
21	Porsche	Editar	Deletar
22	Renault	Editar	Deletar

Fonte: Autoria própria.

A seção de "Gerenciamento de Marcas" (Figura 17) apresenta uma tabela administrativa limpa e organizada. O cabeçalho em azul ciano define as colunas de dados ("ID" e "Nome") e a coluna de operações. As linhas listam as fabricantes cadastradas no sistema, oferecendo controles diretos de "Editar" e "Deletar" para a manutenção rápida dos registros.

Esta interface gerencia uma tabela auxiliar (*lookup table*) do banco de dados, essencial para alimentar os seletores de criação de anúncios. O *backend* realiza uma leitura sequencial dos registros e renderiza a lista via *loop*. Os botões de ação manipulam chaves primárias específicas, enviando parâmetros via *URL* para acionar as rotas de atualização ou remoção da marca alvo.

Figura 18: Gerenciamento de Modelos do sistema Autochase.

Gerenciamento de Modelos

ID	Marca do modelo	Nome do modelo	Fipe	Ações	
1	Toyota	Corolla	R$: 37.898	Editar	Deletar
3	Toyota	Corolla Cross	R$: 300.000	Editar	Deletar
21	Toyota	Yaris	R$: 130.000	Editar	Deletar
22	Toyota	Hilux	R$: 250.000	Editar	Deletar
23	Toyota	SW4	R$: 421.000	Editar	Deletar
24	Toyota	Prius	R$: 168.000	Editar	Deletar
2	Honda	Civic	R$: 190.010	Editar	Deletar
5	Ford	Fusion	R$: 100.000	Editar	Deletar
26	Ford	Escort	R$: 8.800	Editar	Deletar
4	Volkswagen	Gol	R$: 45.000	Editar	Deletar
7	Audi	RS6	R$: 111.111	Editar	Deletar
8	Mazda	RX7	R$: 70.000	Editar	Deletar
10	Chevrolet	Vectra	R$: 29.858	Editar	Deletar
11	Ferrari	458 Italia	R$: 2.600.000	Editar	Deletar
12	Corvette	C8	R$: 1.130.000	Editar	Deletar
18	Porsche	911	R$: 758.000	Editar	Deletar
19	Porsche	718	R$: 550.000	Editar	Deletar
20	Porsche	Cayenne	R$: 505.000	Editar	Deletar
25	Renault	Clio	R$: 200.000	Editar	Deletar

Fonte: Autoria própria.

A seção "Gerenciamento de Modelos" (Figura 18) apresenta uma tabela detalhada que vincula os produtos às suas fabricantes. O cabeçalho azul estrutura as colunas "ID", "Marca do modelo", "Nome do modelo" e o valor de referência "Fipe". As linhas exibem a lista completa de modelos cadastrados, associando visualmente cada carro à sua marca, com os tradicionais controles de "Editar" e "Deletar" disponíveis na coluna final.

Do ponto de vista da arquitetura de dados, esta tela exibe o resultado de uma relação de chave estrangeira (*Foreign Key*). O *backend* executa uma consulta *SQL* com *JOIN* para trazer o nome da marca (armazenado em outra tabela) junto com os dados do modelo. A coluna "Fipe" aplica uma máscara de formatação monetária na renderização, enquanto os botões de ação permitem a gestão do ciclo de vida (*CRUD*) desses registros dependentes.

Figura 19: Gerenciamento de Cores e Gerenciamento de Chassis do sistema Autochase.

Gerenciamento de Cores

ID da cor	Nome da cor	Ações	
1	Preto	Editar	Deletar
2	Branco	Editar	Deletar
3	Prata	Editar	Deletar
4	Vermelho	Editar	Deletar
5	Verde	Editar	Deletar
9	Amarelo	Editar	Deletar
10	Azul	Editar	Deletar
11	Rosa	Editar	Deletar
12	Roxo	Editar	Deletar
14	Cinza	Editar	Deletar

Gerenciamento de Chassis

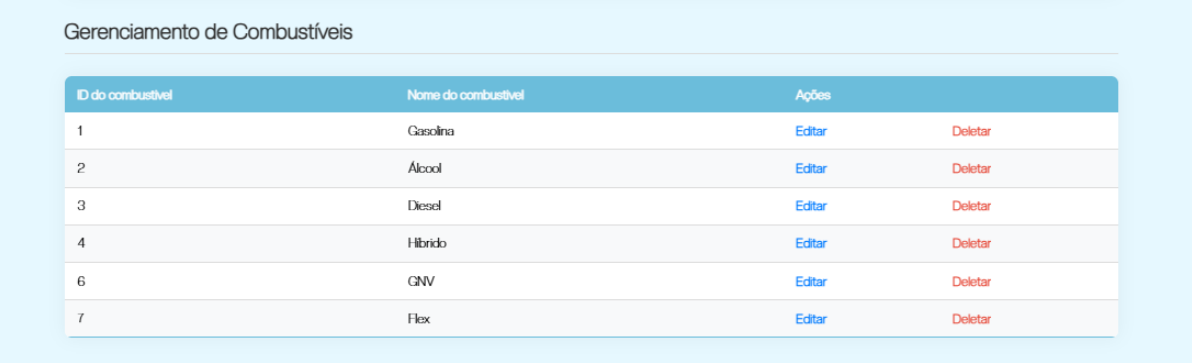
ID da carroceria	Nome da carroceria	Ações	
1	SW	Editar	Deletar
2	Sedan	Editar	Deletar
3	SUV	Editar	Deletar
4	Pickup	Editar	Deletar
6	Coupé	Editar	Deletar
7	Hatchback	Editar	Deletar
8	Shooting Brake	Editar	Deletar
9	Roadster	Editar	Deletar
10	Cabriolet	Editar	Deletar
11	Spyder	Editar	Deletar
12	Barchetta	Editar	Deletar

Fonte: Autoria própria.

A imagem apresenta duas seções administrativas dispostas verticalmente: "Gerenciamento de Cores" e "Gerenciamento de Chassis" (Figura 19). Ambas seguem o padrão visual de tabelas de dados com cabeçalho azul, listando, respectivamente, as opções de pintura (como "Preto", "Branco") e tipos de carroceria (como "Sedan", "SUV") disponíveis no sistema. A coluna final de "Ações" padroniza os controles de manutenção com *links* para "Editar" e "Deletar".

Sob o aspecto técnico, estas tabelas gerenciam as entidades auxiliares (*lookup tables*) do banco de dados, fundamentais para a normalização dos dados e alimentação dos *dropdowns* nos formulários de cadastro. A renderização é feita via laço de repetição (*loop*) no *backend*, e as operações de exclusão ou edição nestes registros refletem imediatamente nas opções disponíveis para os usuários finais, garantindo a consistência das categorias do sistema.

Figura 20: Gerenciamento de Combustíveis do sistema Autochase.



Gerenciamento de Combustíveis

ID do combustível	Nome do combustível	Ações	
1	Gasolina	Editar	Deletar
2	Álcool	Editar	Deletar
3	Diesel	Editar	Deletar
4	Híbrido	Editar	Deletar
6	GNV	Editar	Deletar
7	Flex	Editar	Deletar

Fonte: Autoria própria.

A seção "Gerenciamento de Combustíveis" (Figura 20) exibe uma tabela administrativa padronizada para o controle das matrizes energéticas aceitas pela plataforma. O cabeçalho em azul organiza os dados em colunas de identificação e nome, enquanto as linhas listam os tipos cadastrados, como "Gasolina" e "Diesel". A coluna final centraliza os controles de manutenção, oferecendo *links* diretos para "Editar" ou "Deletar" cada item.

Tecnicamente, esta interface permite a administração das categorias de motorização do sistema. Os dados exibidos alimentam diretamente os *dropdowns* de filtros e cadastros, garantindo a normalização das entradas no banco de dados. A renderização utiliza um laço de repetição (*loop*) no *backend*, onde cada linha representa um registro da tabela auxiliar, manipulável através de requisições disparadas pelos botões de ação para manter a consistência referencial dos anúncios.

Figura 21: Gerenciamento de Anúncios do sistema Autochase.

Gerenciamento de Anúncios

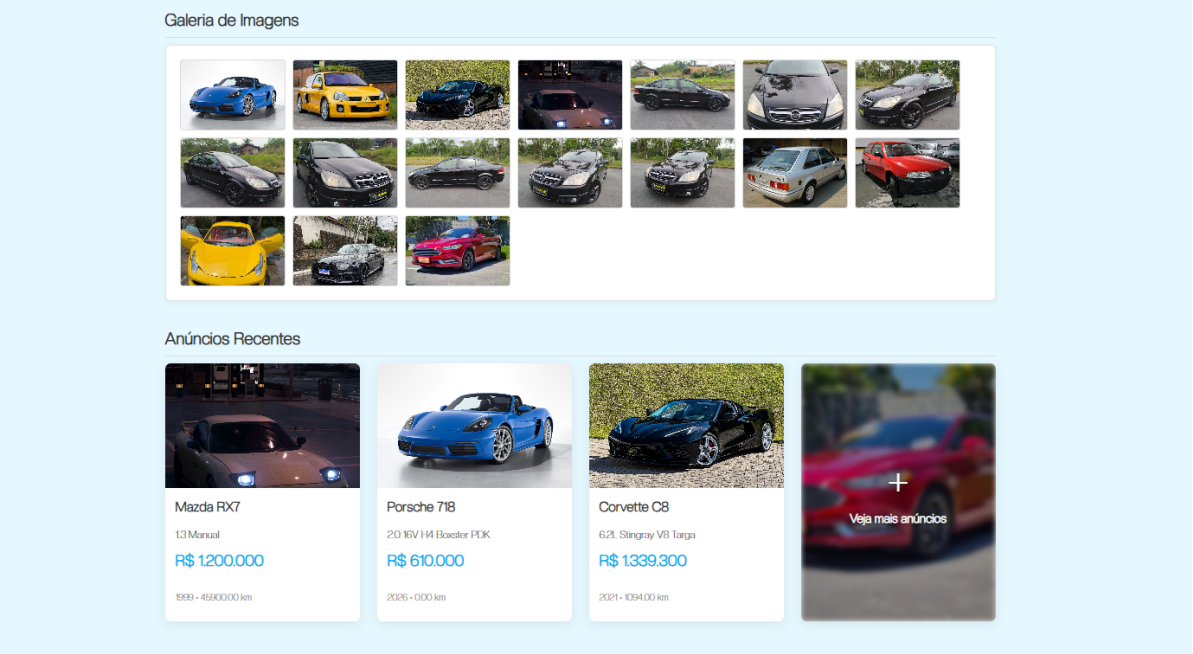
ID do Anúncio	Data de Criação	Descrição	Valor	Marca	Modelo	Versão	Chassi	Combustível	Ações	
36	2025-11-23 03:10:26	bom de motor	R$ 10.400	Ford	Escort	1.0 Hobby	Hatchback	Gasolina	Editar	Deletar
35	2025-11-23 03:07:00	Renault Clio V6 RS super raro	R$ 709.900	Renault	Clio	3.0 V6 RS	Hatchback	Gasolina	Editar	Deletar
31	2025-11-22 02:32:08	inteiro nunca fuçado	R$ 1.200.000	Mazda	RX7	1.3 Manual	Coupé	Gasolina	Editar	Deletar
30	2025-11-21 22:42:54	simples carro porsche 0 km	R$ 610.000	Porsche	718	2.0 16V H4 Boxster PDK	Spyder	Gasolina	Editar	Deletar
29	2025-11-20 23:33:02	Destaques Técnicos: Motor: V8 6.2L LT2	R$ 1.339.300	Corvette	C8	6.2L Stingray V8 Targa	Coupé	Gasolina	Editar	Deletar
27	2025-11-17 20:03:00	Motor: 4.5L V8 "F136" Aspirado, 570cv. Câmbio F1 Dual-Clutch 7	R$ 3.000.000	Ferrari	458 Italia	4.5L V8 Dual-Clutch	Coupé	Gasolina	Editar	Deletar
26	2025-11-17 19:59:29	Golzinho filezinho, pegar e andar. Quem chegar primeiro leva essa reliquia	R$ 10.120	Volkswagen	Gol	1.0 16v G3	Hatchback	Gasolina	Editar	Deletar
25	2025-11-17 01:25:48	Manual F23 - Exclusivo Especial para quem busca conforto,	R$ 50.100	Chevrolet	Vectra	2.4 16v Elite	Sedan	Flex	Editar	Deletar
24	2025-11-16 19:40:34	Veículo com histórico de manutenção rigorosamente em dia na	R$ 899.000	Audi	RS6	4.0 Avant V8 Twinturbo MHEV Tiptronic	SW	Gasolina	Editar	Deletar
23	2025-11-13 00:35:49	GASOLINA 4P AUTOMÁTICO	R$ 90.900	Ford	Fusion	2.0 TITANIUM AWD 16V	Sedan	Híbrido	Editar	Deletar

Fonte: Autoria própria.

A seção "Gerenciamento de Anúncios" (Figura 21) apresenta o inventário completo de vendas em uma tabela densa e detalhada. O cabeçalho azul organiza múltiplas colunas informativas, incluindo "Data de Criação", "Valor" e dados técnicos do veículo. Um destaque visual é a coluna "Descrição", que implementa barras de rolagem verticais internas nas células com muito texto, evitando que a altura da linha se expanda excessivamente. A coluna final de "Ações" mantém os controles padrão de "Editar" e "Deletar".

Tecnicamente, esta tabela consolida a visão administrativa global, exigindo *joins* complexos no *backend* para unificar dados relacionais (como marcas e modelos) em uma única visualização. O diferencial de *frontend* reside na célula de descrição: o uso da propriedade *CSS* “overflow-y: auto” com altura máxima definida cria um *scroll* local, preservando a diagramação da tabela. Os campos de valor e data aplicam máscaras de formatação visual, enquanto os botões de ação gerenciam o ciclo de vida do anúncio via parâmetros *GET*.

Figura 22: Galeria de Imagens e Anúncios Recentes do sistema Autochase.



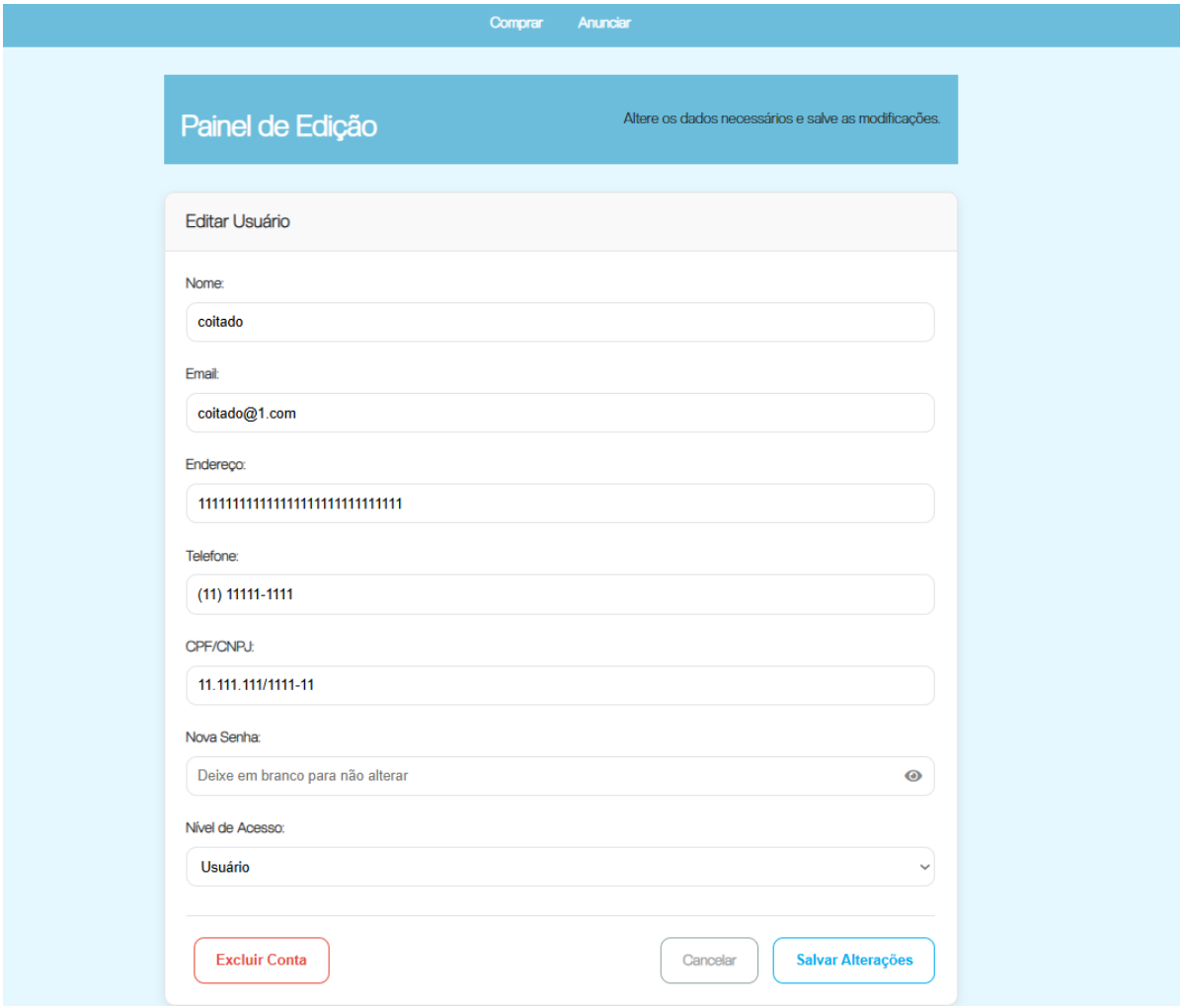
Fonte: Autoria própria.

A interface apresenta uma seção superior denominada "Galeria de Imagens" (Figura 22), contida em um painel branco que organiza visualmente o acervo de fotos do sistema. As miniaturas são dispostas em uma grade compacta, permitindo a visualização simultânea de múltiplos veículos em diversos ângulos. Logo abaixo, a estrutura conecta-se à área de "Anúncios Recentes", exibindo os cartões de produtos padrão com destaque para modelos como "Mazda RX7" e "Porsche 718", seguidos pelo cartão de navegação que convida o usuário a ver mais ofertas.

Tecnicamente, a galeria implementa uma renderização de coleção de mídia. O *backend* recupera os caminhos dos arquivos armazenados no servidor e os injeta no *DOM* através de uma estrutura de repetição (*loop*). O *frontend* utiliza *CSS Grid* ou *Flexbox* com a propriedade *flex-wrap* para organizar os itens, enquanto as imagens aplicam *object-fit: cover* para preencher uniformemente as células da grade, garantindo que fotos de diferentes proporções (paisagem ou retrato) mantenham a harmonia visual sem distorções.

8.11. Tela de Edições do Painel de Administrador

Figura 23: Edição de usuário do Administrador do sistema Autochase.



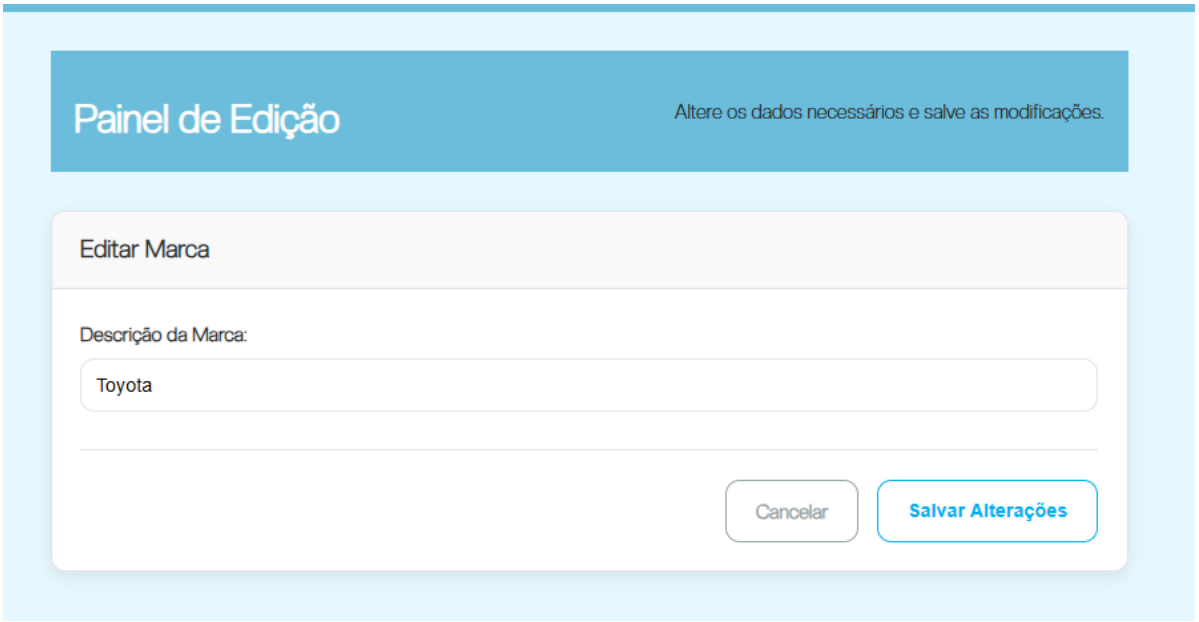
Fonte: Autoria própria.

A interface de "Edição de Usuário via Admin" (Figura 23) reaproveita integralmente o *layout* e a estrutura visual apresentada na Tela de Edição do Usuário (Figura 7). A disposição dos campos de dados pessoais, como "Nome", "Email" e "CPF/CNPJ", permanece idêntica, mantendo a consistência visual do sistema. Contudo, a diferença crucial neste contexto administrativo reside na interatividade do

campo "Nível de Acesso". Enquanto na visão do usuário comum este dado é geralmente estático ou apenas informativo, nesta tela ele se apresenta como um seletor ativo (*dropdown*), permitindo ao administrador redefinir a hierarquia da conta alvo.

Tecnicamente, embora compartilhe a mesma *view* da Tela de Edição do Usuário (Figura 7), a lógica de *backend* operante aqui é distinta. A consulta de carregamento não utiliza o *ID* da sessão do usuário logado, mas sim o identificador capturado via parâmetro *GET* da tabela de gerenciamento. A funcionalidade crítica é a manipulação do *RBAC* (*Role-Based Access Control*): ao submeter este formulário, o sistema permite a elevação ou revogação de privilégios (alternando entre "USUARIO" e "ADMIN"), uma operação de alto risco que exige validação rigorosa de segurança no servidor para evitar escalonamento de privilégios indevido.

Figura 24: Edição de Marca do Administrador do sistema Autochase.



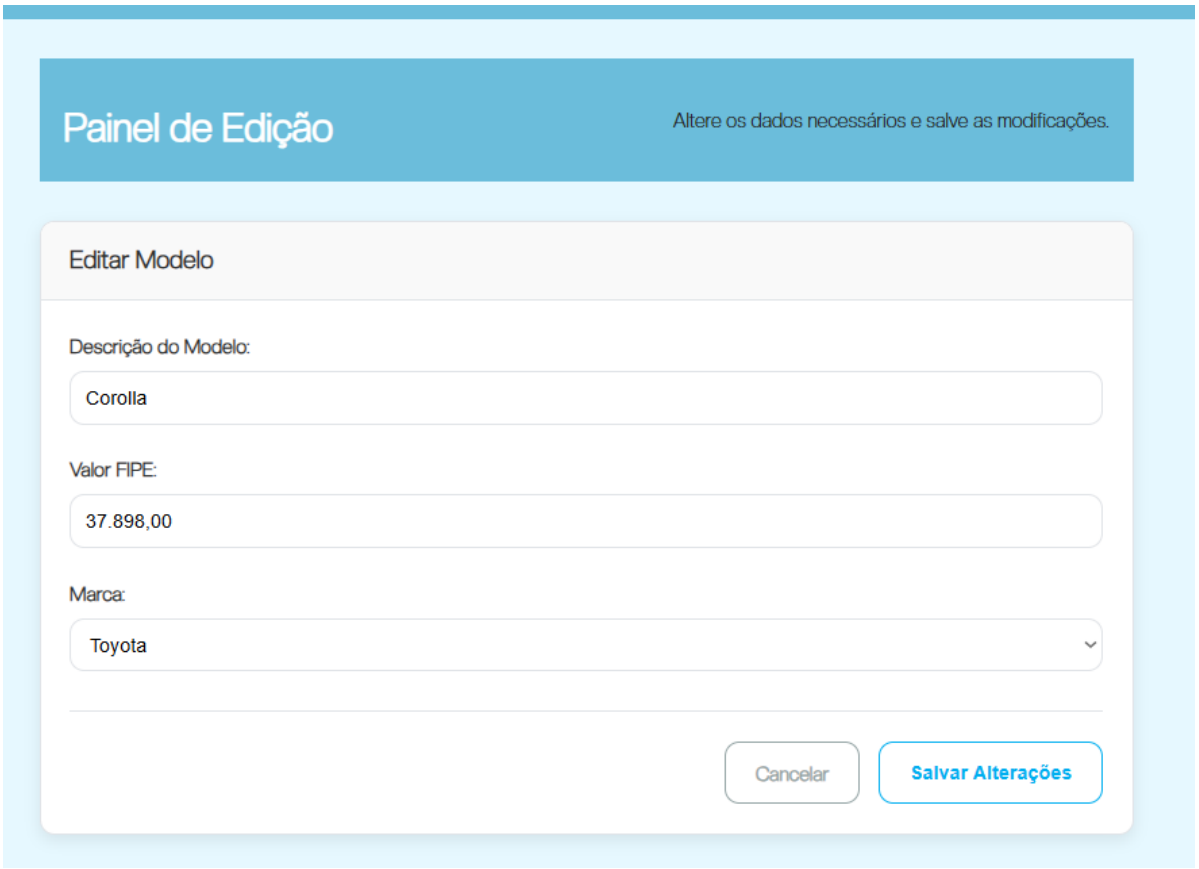
Fonte: Autoria própria.

A interface de "Painel de Edição" (Figura 24) foca aqui na manutenção de um dado específico. O formulário é apresentado em um cartão centralizado sob o título "Editar Marca", contendo um único campo de texto rotulado como "Descrição da

Marca". Este campo já vem preenchido com o valor atual (ex: "Toyota"), permitindo a correção imediata. O rodapé do cartão mantém o padrão de navegação do sistema com os botões "Cancelar" e "Salvar Alterações".

Tecnicamente, esta tela executa a operação atômica de *Update* em uma entidade simples do banco de dados. O *input* é populado via *query SQL* baseada no *ID* recebido pela *URL*. Ao confirmar a ação, o *backend* dispara um comando *UPDATE* na tabela de marcas. Esta alteração é crítica pois reflete em cascata: a correção de um nome aqui atualiza instantaneamente todos os *dropdowns* de filtros e formulários de cadastro que referenciam esta marca, garantindo a integridade dos dados exibidos.

Figura 25: Edição de Modelo do Administrador do sistema Autochase.



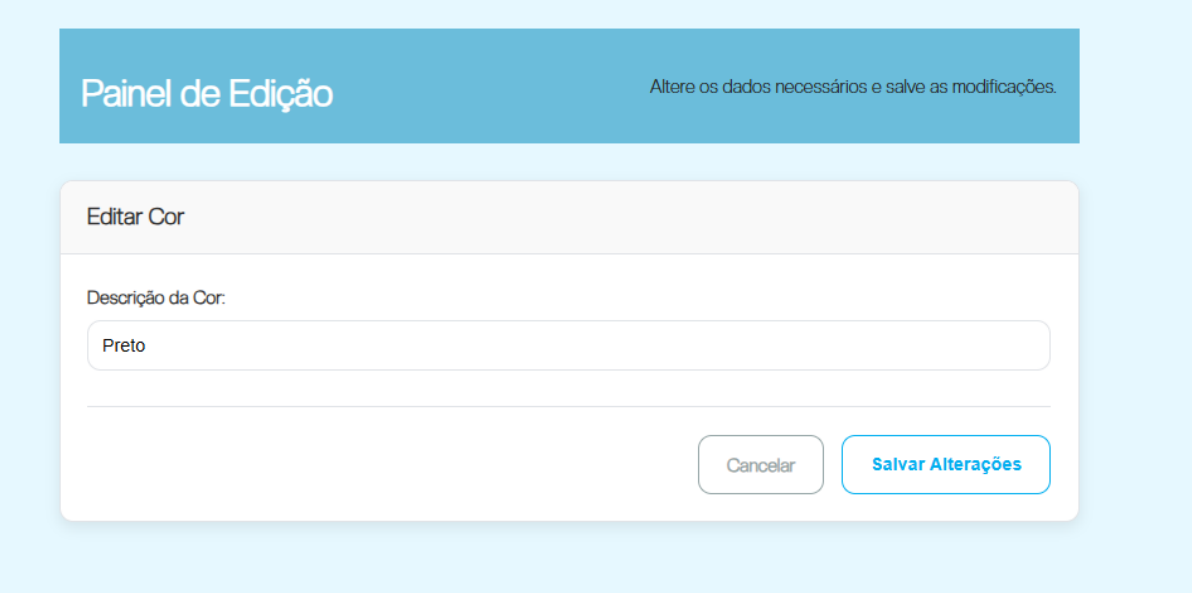
Fonte: Autoria própria.

A tela "Painel de Edição" (Figura 25) apresenta aqui um formulário dedicado à atualização de modelos de veículos. O cartão "Editar Modelo" estrutura três campos

essenciais: uma caixa de texto para a "Descrição do Modelo", um campo numérico formatado para o "Valor FIPE" e um menu de seleção (*dropdown*) para a "Marca". Esta disposição permite ao administrador não apenas corrigir o nome e o valor de mercado, mas também redefinir a qual fabricante aquele modelo pertence, finalizando o processo com os botões "Cancelar" ou "Salvar Alterações".

Tecnicamente, este formulário manipula uma entidade que possui dependência relacional. O campo "Marca" visualiza a chave estrangeira (*Foreign Key*) da tabela de modelos, permitindo a alteração do vínculo com a tabela de fabricantes via *select* HTML. Já o campo "Valor FIPE" aplica uma máscara de *input* no *frontend* para exibir a pontuação e vírgula monetária, exigindo que o *backend* converta essa *string* formatada de volta para *decimal* ou *float* antes de persistir o comando *UPDATE* no banco de dados.

Figura 26: Edição de Cor do Administrador do sistema Autochase.

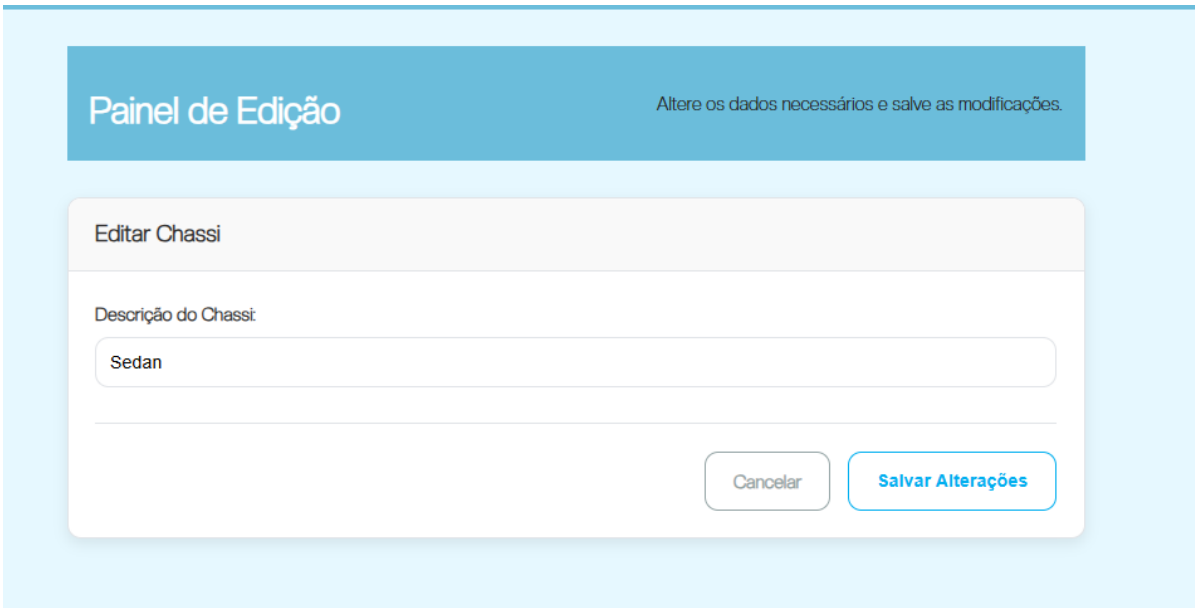


Fonte: Autoria própria.

A interface "Painel de Edição" (Figura 26) é aplicada aqui para a manutenção de atributos básicos. O cartão central "Editar Cor" apresenta um *layout* minimalista contendo apenas o campo "Descrição da Cor", preenchido previamente com o dado atual (ex: "Preto"). A interação é concluída através dos botões de controle "Cancelar" e "Salvar Alterações", mantendo a coerência visual das demais telas administrativas.

Tecnicamente, esta tela realiza a atualização de uma tabela auxiliar do banco de dados. O formulário recebe o valor atual via consulta baseada no *ID* da cor e, após a submissão, o *backend* executa um comando *UPDATE* simples. Esta alteração reflete imediatamente em todo o sistema, atualizando os filtros e detalhes de anúncios que utilizam essa chave estrangeira para exibir a característica do veículo.

Figura 27: Edição de Chassi do Administrador do sistema Autochase.

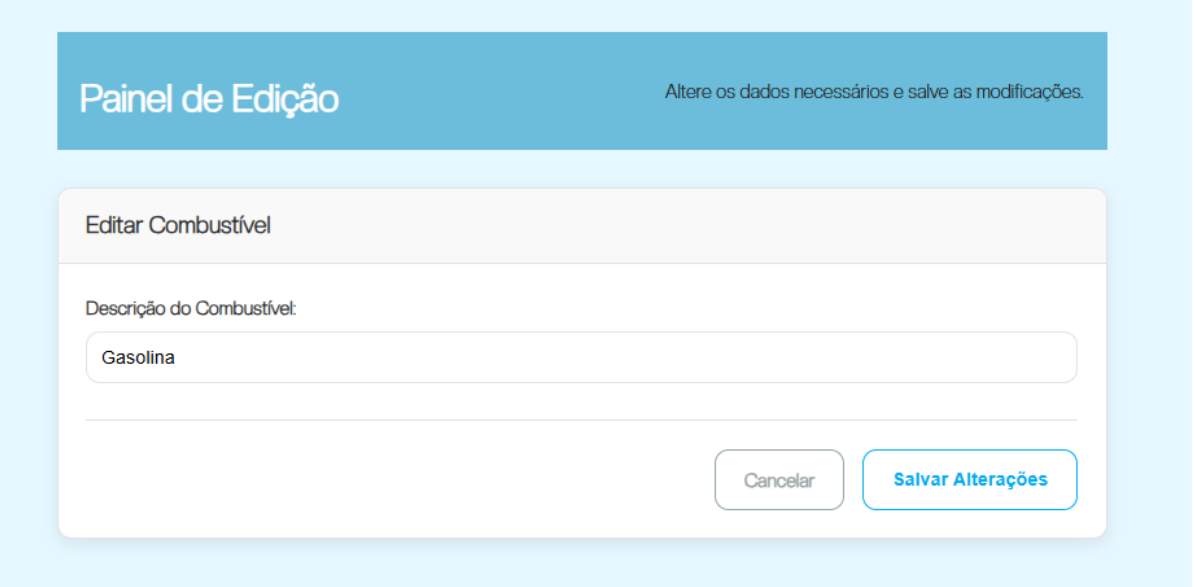


Fonte: Autoria própria.

A interface de "Painel de Edição" (Figura 27) é utilizada aqui para a atualização das categorias de carroceria. O cartão centralizado, intitulado "Editar Chassi", exibe um campo de texto único sob o rótulo "Descrição do Chassi", contendo o dado a ser modificado (ex: "Sedan"). O fluxo de interação é encerrado pelos botões de comando padrão "Cancelar" e "Salvar Alterações" no rodapé do componente.

Sob a ótica do desenvolvimento, este formulário gerencia a persistência de dados na tabela de tipos de carroceria. O carregamento da página popula o *input* com o valor recuperado do banco de dados através de uma requisição *GET*. A submissão do formulário aciona o método de *Update* no *backend*, corrigindo a nomenclatura do chassi e refletindo a mudança instantaneamente em todos os seletores de filtros e cadastros que consomem essa lista.

Figura 28: Edição de Combustível do Administrador do sistema Autochase.



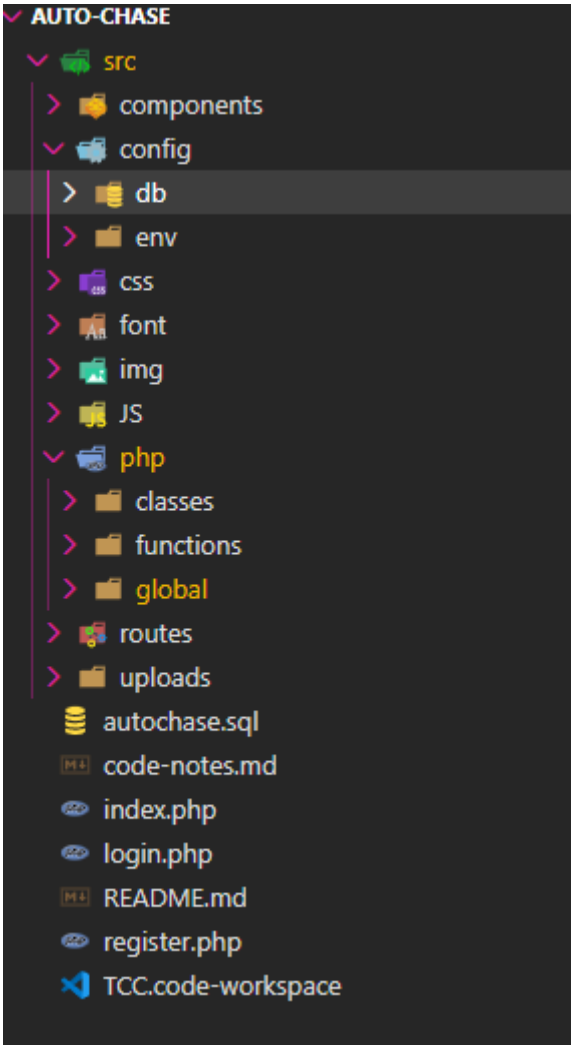
Fonte: Autoria própria.

A interface de "Painel de Edição" (Figura 28) é aplicada aqui para a manutenção das categorias de energia. O formulário central, "Editar Combustível", exibe um *layout* limpo com um único campo de texto para a "Descrição do Combustível", preenchido com o dado vigente (ex: "Gasolina"). A interação é finalizada pelos botões de ação "Cancelar" e "Salvar Alterações", seguindo a consistência visual do módulo administrativo.

Do ponto de vista técnico, este formulário gerencia a atualização da tabela auxiliar de combustíveis. O *input* é carregado com o valor recuperado do banco de dados via requisição *GET*. Ao submeter a alteração, o *backend* processa um comando *UPDATE*, garantindo que a correção da nomenclatura reflita instantaneamente nos filtros de busca e nos formulários de criação de anúncios que dependem dessa classificação.

8.12. Estrutura de pastas do sistema

Figura 30: Estrutura de pastas do sistema Autochase.



Fonte: Autoria própria.

8.13. Camada de configuração

Esta pasta isola as configurações globais do ambiente, garantindo segurança e reutilização de código.

- **db/**: Contém o arquivo connect.php, responsável exclusivamente por estabelecer a conexão com o banco de dados MySQL via PDO ou MySQLi. Isso centraliza as credenciais de acesso, facilitando a manutenção.
- **env/**: Contém scripts de ambiente e controle de sessão, como logout.php e

importações globais, gerenciando o ciclo de vida do usuário no sistema.

8.14. Camada de Modelo e Lógica de Negócio (php/)

É o núcleo do processamento do sistema ("Core"), subdividido para melhor organização:

- **classes/**: Implementa o paradigma de Orientação a Objetos. Cada arquivo (ex: class-anuncio.php, class-usuario.php, class-veiculo.php) representa uma entidade do banco de dados. Essas classes contêm os métodos para manipulação de dados (CRUD - Create, Read, Update, Delete) e encapsulam as regras de negócio.
- **functions/**: Contém bibliotecas de funções auxiliares e procedurais que dão suporte ao sistema, como filtros de busca (filter-functions.php) e rotas para requisições AJAX (modelo_json.php), permitindo carregamento dinâmico de dados sem recarregar a página.
- **routes/**: Atua como a camada de controle de navegação. Arquivos como anuncio.php, compra.php e meus-anuncios.php gerenciam o fluxo das páginas específicas, recebendo as requisições do usuário e conectando as classes de lógica à interface visual.

8.15. Apresentação e Estilo (css/, font/, img/, js/)

Responsável pela interface gráfica (UI) e experiência do usuário (UX).

- **css/:** Diferente de abordagens monolíticas, o CSS foi modularizado por contexto. Existem folhas de estilo específicas para o painel administrativo (adminpanel.css), cartões de produtos (card.css), login e registro. Isso evita conflitos de estilo e mantém o código limpo.
- **font/**: O sistema utiliza tipografia personalizada (família CreatoDisplay), armazenada localmente para garantir a identidade visual consistente independente do dispositivo do usuário.
- **js/:** Contém a lógica client-side (JavaScript), como o arquivo js-functions.js, responsável por validação de formulário no navegador e interatividade da interface.
- **components/**: Armazena fragmentos de código reutilizáveis que se

repetem em múltiplas páginas, como o rodapé (footer.php)

8.16. Gerenciamento de Arquivos (uploads/)

Diretório dedicado ao armazenamento de conteúdo gerado pelo usuário (User Generated Content), especificamente as imagens dos veículos cadastrados nos anúncios. A separação desta pasta facilita rotinas de backup e previne a mistura de arquivos de código-fonte com arquivos de mídia.

- **uploads/:** Contém as imagens geradas pelo usuários.

8.17. Arquivos na Raiz

Na raiz do projeto, encontram-se os pontos de entrada principais:

- **index.php**: A home page é o ponto de partida da navegação.
- **login.php / register.php**: Interfaces de acesso e cadastro.

9. CONSIDERAÇÕES FINAIS

O desenvolvimento do sistema AutoChase permitiu alcançar satisfatoriamente os objetivos propostos, resultando na entrega de uma plataforma *web* funcional para a intermediação de compra e venda de veículos. Trabalhar neste projeto revelou-se uma experiência extremamente gratificante, possibilitando observar na prática a orquestração completa entre o banco de dados, o *backend* e a interface (*frontend*). A aplicação foi capaz de implementar com êxito os requisitos fundamentais de gerenciamento de anúncios, validando a arquitetura *MVC* adaptada e a estruturação do banco de dados relacional.

Ao longo do cronograma, foi possível notar uma evolução nítida do *software*: a cada nova modificação, o projeto foi lapidado e as arestas eliminadas, transformando-se gradualmente em uma obra bem acabada. O êxito principal reside na criação de um ambiente seguro e normalizado, onde a integridade das

informações é assegurada através de regras de negócio estritas — como a criação atômica de veículos e anúncios e o controle de acesso baseado em níveis hierárquicos — garantindo que a manipulação de dados sensíveis fique restrita ao perfil administrativo. Esteticamente, houve uma harmonia significativa entre as escolhas de *design* e a execução técnica; o resultado final apresenta-se visualmente coeso e organizado, refletindo a qualidade tanto na estrutura do código quanto na interface do usuário.

Durante o percurso de desenvolvimento, as principais dificuldades encontradas estiveram relacionadas à opção deliberada pelo uso de tecnologias nativas (*PHP* puro e *JavaScript* sem *frameworks*). A ausência de bibliotecas de abstração exigiu um esforço significativo na implementação manual de funcionalidades como o sistema de roteamento, a gestão de sessões e a segurança contra injeção de *SQL*. Além disso, a lógica de transação para garantir o cadastro simultâneo de veículo e anúncio demandou um estudo aprofundado sobre o controle de transações no *MySQL*, representando um desafio técnico considerável para assegurar a consistência dos registros.

Entretanto, esses desafios converteram-se nas principais contribuições para o desenvolvimento acadêmico e profissional. A construção do sistema "do zero" proporcionou um entendimento profundo sobre o funcionamento real do protocolo *HTTP*, o ciclo de vida das requisições e a importância da Programação Orientada a Objetos. O domínio adquirido sobre a manipulação direta do *DOM* com *JavaScript* e a estruturação de consultas *SQL* complexas consolidou uma base técnica robusta, preparando os desenvolvedores para compreender a engenharia de software que sustenta as ferramentas de mercado. O projeto serviu como uma consolidação moral da capacidade de entrega, evidenciando a importância da normalização de dados e da organização do código.

Vislumbrando possibilidades de trabalhos futuros, o sistema AutoChase apresenta uma base sólida para diversas expansões. Embora o projeto tenha sido finalizado com sucesso, há pontos visíveis de melhoria, como a estabilização de uma versão *mobile* responsiva. Evoluções naturais incluiriam a implementação de uma *API RESTful* para permitir aplicativos nativos e a atualização do módulo de *chat* para comunicação em tempo real via *WebSockets*. Por fim, a integração com

gateways de pagamento digital para transações financeiras agregaria valor comercial, transformando este sistema acadêmico em uma solução de mercado competitiva.

10. REFERÊNCIAS

IBISWORLD. *Global car & automobile sales industry report*. 2023.

W3SCHOOLS. *PHP tutorial*. Disponível em: https://www.w3schools.com/php/default.asp. Acesso em: 24 nov. 2025.

W3SCHOOLS. *PHP: MySQL database*. Disponível em: https://www.w3schools.com/php/php_mysql_intro.asp. Acesso em: 24 nov. 2025.

W3SCHOOLS. *PHP – AJAX introduction*. Disponível em: https://www.w3schools.com/php/php_ajax_intro.asp. Acesso em: 24 nov. 2025.

HEUSER, C. A. **Projeto de banco de dados**. 6. ed. Porto Alegre: Bookman, 2009. Disponível em: https://apnp.ifsul.edu.br/mod/resource/view.php?id=1162333

https://apnp.ifsul.edu.br/mod/resource/view.php?id=1162345

https://apnp.ifsul.edu.br/mod/resource/view.php?id=1162353. Acesso em: 30 Out. 2024.

BROD, F. A. T. **Banco de Dados I: Modelagem Conceitual - Modelo de Entidade e Relacionamento**. Slides. Disciplina de Banco de Dados I – Curso Técnico em Desenvolvimento de Sistemas. Instituto Federal Sul-rio-grandense. Disponível em: https://apnp.ifsul.edu.br/mod/resource/view.php?id=1162334. Acesso em: 9 Out. 2024.

MOREIRA, M. I. G. **Análise e Projeto de Sistemas: Aula 06 - Diagrama de Caso de Uso**. Slides. Disciplina de Análise e Projeto de Sistemas – Curso Técnico em

Desenvolvimento de Sistemas. Instituto Federal Sul-rio-grandense. Disponível em: https://apnp.ifsul.edu.br/mod/resource/view.php?id=1162486. Acesso em: 6 Nov. 2024

LOPES, J. L. B. **Banco de Dados II: phpMyAdmin Visão Geral**. Slides. Disciplina de Banco de Dados II – Curso Técnico em Desenvolvimento de Sistemas. Instituto Federal Sul-rio-grandense. Disponível em: https://apnp.ifsul.edu.br/mod/resource/view.php?id=1299804. Acesso em: 25 Mar. 2025.

RODRIGUES, A. S. **Linguagem de Programação II: Aula 9 - PHP Conexão com Banco de Dados**. Slides. Disciplina de Linguagem de Programação II – Curso Técnico em Desenvolvimento de Sistemas. Instituto Federal Sul-rio-grandense. Disponível em: https://apnp.ifsul.edu.br/mod/resource/view.php?id=1300164. Acesso em: 7 Abr. 2025.

RODRIGUES, A. S. **Linguagem de Programação II: PHP - SESSION - Variáveis de Sessão e Ferramenta de Login**. Slides. Disciplina de Linguagem de Programação II – Curso Técnico em Desenvolvimento de Sistemas. Instituto Federal Sul-rio-grandense. Disponível em: https://apnp.ifsul.edu.br/mod/resource/view.php?id=1300188. Acesso em: 2 Jun. 2025.

RODRIGUES, A. S. **Linguagem de Programação II: Java Script - DOM (Document Object Model)**. Slides. Disciplina de Linguagem de Programação II – Curso Técnico em Desenvolvimento de Sistemas. Instituto Federal Sul-rio-grandense. Disponível em: https://apnp.ifsul.edu.br/mod/resource/view.php?id=1402883. Acesso em: 16 Jun. 2025.

INSTITUTO FEDERAL SUL-RIO-GRANDENSE. **Estrutura do TCC 2: Curso Técnico em Desenvolvimento de Sistemas**. Documento de orientação acadêmica. Pelotas: Câmpus Visconde da Graça. 2025. Disponível em: https://apnp.ifsul.edu.br/mod/resource/view.php?id=1425023.

APÊNDICE I - Instruções SQL para Criação da Base de Dados

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Tempo de geração: 24/11/2025 às 19:30
-- Versão do servidor: 10.4.32-MariaDB
-- Versão do PHP: 8.2.12

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Banco de dados: `auto-chase`
--

-- --------------------------------------------------------

--
-- Estrutura para tabela `anuncio`
--

CREATE TABLE `anuncio` (
  `anuncio_id` int(11) NOT NULL,
  `anuncio_valor` decimal(10,2) NOT NULL,
  `anuncio_data_de_alteracao` timestamp NOT NULL DEFAULT current_timestamp()
ON UPDATE current_timestamp(),
  `anuncio_data_de_criacao` timestamp NOT NULL DEFAULT current_timestamp(),
  `anuncio_desc` varchar(2000) DEFAULT NULL,
  `anuncio_status` enum('ATIVO','INATIVO','VENDIDO') NOT NULL DEFAULT
'ATIVO',
```

```
  `fk_usuario_id` int(11) NOT NULL,
  `fk_veiculo_id` int(11) NOT NULL COMMENT 'FK do veículo'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Despejando dados para a tabela `anuncio`
--

INSERT INTO `anuncio` (`anuncio_id`, `anuncio_valor`,
`anuncio_data_de_alteracao`, `anuncio_data_de_criacao`, `anuncio_desc`,
`anuncio_status`, `fk_usuario_id`, `fk_veiculo_id`) VALUES
(23, 90900.00, '2025-11-22 04:03:17', '2025-11-13 03:35:49', 'GASOLINA 4P
AUTOMÁTICO\r\n', 'ATIVO', 5, 25),
(24, 899000.00, '2025-11-21 09:26:38', '2025-11-16 22:40:34', 'Veículo com histórico
de manutenção rigorosamente em dia na concessionária. Sem retoques, sem stage
(mapa), totalmente original. Para quem busca exclusividade e não aceita andar
atrás.', 'ATIVO', 5, 26),
(25, 50100.00, '2025-11-23 06:25:27', '2025-11-17 04:25:48', 'Manual F23 -
Exclusivo\r\nEspecial para quem busca conforto, desempenho para viagens e bom
custo-benefício no mercado de usados. Seus pontos positivos incluem motor forte e
suave, acabamento interno luxuoso e suspensão confortável.', 'ATIVO', 5, 27),
(26, 10120.00, '2025-11-21 08:43:35', '2025-11-17 22:59:29', 'Golzinho filezinho,
pegar e andar. Quem chegar primeiro leva essa relíquia', 'ATIVO', 9, 28),
(27, 3000000.00, '2025-11-21 08:27:42', '2025-11-17 23:03:00', 'Motor: 4.5L V8
\"F136\" Aspirado, 570cv.\r\nCâmbio F1 Dual-Clutch 7 marchas, Interior com
acabamentos em fibra de carbono.\r\nVolante em Carbono + LEDs, Opcional
indispensável para a experiência de pilotagem.\r\nEscudos Scuderia Ferrari
Embutidos nos para-lamas.\r\nFreios: Carbono-Cerâmica com pinças Giallo
Modena.\r\nInterior Couro de alta gramatura em estado de conservação impecável,
sem desgastes.\r\nVeículo mantido com rigor técnico absoluto. Revisões em dia.
Carro para colecionadores que entendem que a era dos motores aspirados de alto
giro acabou.', 'ATIVO', 5, 29),
(29, 1339300.00, '2025-11-21 06:51:06', '2025-11-21 02:33:02', 'Destaques
Técnicos:\r\n\r\nMotor: V8 6.2L LT2 Aspirado | 495cv (Z51 Package).\r\n\r\nCâmbio:
Tremec DCT 8 Velocidades (Dupla Embreagem).\r\n\r\nPerformance: 0 a 100 km/h
em 2,9s.\r\n\r\nConfiguração & Opcionais:\r\n\r\nPacote Z51 Performance: Freios
Brembo maiores, diferencial eLSD, escape esportivo e refrigeração
aprimorada.\r\n\r\nFront Lift: Sistema de elevação de eixo dianteiro com memória
GPS (item obrigatório para o Brasil).\r\n\r\nInterior 3LT: Acabamento máximo em
couro Napa, bancos GT2 em fibra de carbono e teto em Alcantara.\r\n\r\nSom: Bose
Performance Series de 14 alto-falantes.\r\n\r\nSuspensão: Magnetic Ride Control
4.0.\r\n\r\nVeículo sem detalhes, com vitrificação de pintura realizada.
Documentação 100% em dia. Apenas venda.', 'ATIVO', 5, 31),
(30, 610000.00, '2025-11-22 01:42:54', '2025-11-22 01:42:54', 'simples carro porsche
```

```
0 km', 'ATIVO', 9, 32),
(31, 1200000.00, '2025-11-23 06:04:28', '2025-11-22 05:32:08', 'inteiro nunca
fuçado', 'ATIVO', 5, 33),
(35, 709900.00, '2025-11-23 06:11:29', '2025-11-23 06:07:00', 'Renault Clio V6 RS
super raro', 'INATIVO', 5, 37),
(36, 10400.00, '2025-11-23 06:11:43', '2025-11-23 06:10:26', 'bom de motor',
'VENDIDO', 5, 38);

-- --------------------------------------------------------

--
-- Estrutura para tabela `chassi`
--

CREATE TABLE `chassi` (
  `chassi_id` int(11) NOT NULL,
  `chassi_desc` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Despejando dados para a tabela `chassi`
--

INSERT INTO `chassi` (`chassi_id`, `chassi_desc`) VALUES
(1, 'SW'),
(2, 'Sedan'),
(3, 'SUV'),
(4, 'Pickup'),
(6, 'Coupé'),
(7, 'Hatchback'),
(8, 'Shooting Brake'),
(9, 'Roadster'),
(10, 'Cabriolet'),
(11, 'Spyder'),
(12, 'Barchetta');

-- --------------------------------------------------------

--
-- Estrutura para tabela `combustivel`
--

CREATE TABLE `combustivel` (
  `comb_id` int(11) NOT NULL,
```

```
  `comb_desc` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Despejando dados para a tabela `combustivel`
--

INSERT INTO `combustivel` (`comb_id`, `comb_desc`) VALUES
(1, 'Gasolina'),
(2, 'Álcool'),
(3, 'Diesel'),
(4, 'Híbrido'),
(6, 'GNV'),
(7, 'Flex');

-- --------------------------------------------------------

--
-- Estrutura para tabela `cor`
--

CREATE TABLE `cor` (
  `cor_id` int(11) NOT NULL,
  `cor_desc` varchar(55) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Despejando dados para a tabela `cor`
--

INSERT INTO `cor` (`cor_id`, `cor_desc`) VALUES
(1, 'Preto'),
(2, 'Branco'),
(3, 'Prata'),
(4, 'Vermelho'),
(5, 'Verde'),
(9, 'Amarelo'),
(10, 'Azul'),
(11, 'Rosa'),
(12, 'Roxo'),
(14, 'Cinza');

-- --------------------------------------------------------
```

```
--
-- Estrutura para tabela `imagem`
--

CREATE TABLE `imagem` (
  `imagem_id` int(11) NOT NULL,
  `imagem_url` varchar(255) NOT NULL,
  `fk_anuncio_id` int(11) NOT NULL,
  `imagem_data_insercao` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Despejando dados para a tabela `imagem`
--

INSERT INTO `imagem` (`imagem_id`, `imagem_url`, `fk_anuncio_id`,
`imagem_data_insercao`) VALUES
(31, 'src/uploads/Jason_and_Lucia_01_landscape.jpg', 23, '2025-11-13 03:35:49'),
(32, 'src/uploads/G54VbpYW8AAC9FU.jpg', 24, '2025-11-16 22:40:34'),
(34, 'src/uploads/G2C7Ne9XIAAFHKB.jpg', 26, '2025-11-17 22:59:29'),
(35, 'src/uploads/G54dihzXIAAS7_R.jpg', 27, '2025-11-17 23:03:00'),
(40, 'src/uploads/chevrolet-corvette---2021-19315689-003.webp', 29, '2025-11-21
02:33:02'),
(41, 'src/uploads/718 Boxster (2024).webp', 30, '2025-11-22 01:42:54'),
(53, 'src/uploads/dcx2omb-46a67bd1-e35d-4212-90cf-5f88bb6ac094.jpg', 31,
'2025-11-23 02:07:46'),
(54, 'src/uploads/9uvgylq6ezqwgqh21rflaij38.jpg', 35, '2025-11-23 06:07:00'),
(55,
'src/uploads/escort-hobby-2p1-7262ce547815a94b2e15614879102516-1024-1024.jp
g', 36, '2025-11-23 06:10:26'),
(88,
'src/uploads/D_NQ_NP_2X_859153-MLB97216360993_112025-F-vectra-elite-24-mp
fi-16v-flexpower-aut.webp', 25, '2025-11-23 06:18:09'),
(89,
'src/uploads/D_NQ_NP_2X_610375-MLB97216360983_112025-F-vectra-elite-24-mp
fi-16v-flexpower-aut.webp', 25, '2025-11-23 06:18:09'),
(90,
'src/uploads/D_NQ_NP_2X_775544-MLB97216666327_112025-F-vectra-elite-24-mp
fi-16v-flexpower-aut.webp', 25, '2025-11-23 06:18:09'),
(91,
'src/uploads/D_NQ_NP_2X_692995-MLB96759583292_112025-F-vectra-elite-24-mp
fi-16v-flexpower-aut.webp', 25, '2025-11-23 06:18:09'),
(92,
'src/uploads/D_NQ_NP_2X_951733-MLB97216360987_112025-F-vectra-elite-24-mp
```

```
fi-16v-flexpower-aut.webp', 25, '2025-11-23 06:18:09'),
(93,
'src/uploads/D_NQ_NP_2X_759109-MLB96761635836_112025-F-vectra-elite-24-mp
fi-16v-flexpower-aut.webp', 25, '2025-11-23 06:18:09'),
(94,
'src/uploads/D_NQ_NP_2X_909480-MLB97216360985_112025-F-vectra-elite-24-mp
fi-16v-flexpower-aut.webp', 25, '2025-11-23 06:18:09'),
(95,
'src/uploads/D_NQ_NP_2X_798131-MLB97216666515_112025-F-vectra-elite-24-mp
fi-16v-flexpower-aut.webp', 25, '2025-11-23 06:18:09');

-- --------------------------------------------------------

--
-- Estrutura para tabela `marca`
--

CREATE TABLE `marca` (
  `marca_id` int(11) NOT NULL,
  `marca_desc` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Despejando dados para a tabela `marca`
--

INSERT INTO `marca` (`marca_id`, `marca_desc`) VALUES
(1, 'Toyota'),
(2, 'Honda'),
(3, 'Ford'),
(4, 'Volkswagen'),
(9, 'Audi'),
(10, 'Mazda '),
(15, 'Chevrolet'),
(17, 'Ferrari'),
(18, 'Corvette'),
(21, 'Porsche'),
(22, 'Renault');

-- --------------------------------------------------------

--
-- Estrutura para tabela `modelo`
--
```

```
CREATE TABLE `modelo` (
  `modelo_id` int(11) NOT NULL,
  `modelo_desc` varchar(55) NOT NULL,
  `modelo_valor_fipe` decimal(10,2) NOT NULL,
  `fk_Marca_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Despejando dados para a tabela `modelo`
--

INSERT INTO `modelo` (`modelo_id`, `modelo_desc`, `modelo_valor_fipe`,
`fk_Marca_id`) VALUES
(1, 'Corolla', 37898.00, 1),
(2, 'Civic', 190010.00, 2),
(3, 'Corolla Cross', 300000.00, 1),
(4, 'Gol', 45000.00, 4),
(5, 'Fusion', 100000.00, 3),
(7, 'RS6', 111111.00, 9),
(8, 'RX7', 70000.00, 10),
(10, 'Vectra', 29857.62, 15),
(11, '458 Italia', 2600000.00, 17),
(12, 'C8', 1130000.00, 18),
(18, '911', 758000.00, 21),
(19, '718', 550000.00, 21),
(20, 'Cayenne', 505000.00, 21),
(21, 'Yaris', 130000.00, 1),
(22, 'Hilux', 250000.00, 1),
(23, 'SW4', 421000.00, 1),
(24, 'Prius', 168000.00, 1),
(25, 'Clio', 200000.00, 22),
(26, 'Escort', 8800.00, 3);

-- --------------------------------------------------------

--
-- Estrutura para tabela `usuario`
--

CREATE TABLE `usuario` (
  `usuario_id` int(11) NOT NULL,
  `usuario_nome` varchar(30) NOT NULL,
  `usuario_senha` varchar(255) NOT NULL,
```

```
  `usuario_nivel_de_acesso` enum('ADMIN','USUARIO') NOT NULL DEFAULT
'USUARIO',
  `usuario_email` varchar(55) NOT NULL,
  `usuario_telefone` varchar(20) NOT NULL,
  `usuario_endereco` varchar(50) NOT NULL,
  `usuario_doc_cpf_cnpj` varchar(25) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Despejando dados para a tabela `usuario`
--

INSERT INTO `usuario` (`usuario_id`, `usuario_nome`, `usuario_senha`,
`usuario_nivel_de_acesso`, `usuario_email`, `usuario_telefone`, `usuario_endereco`,
`usuario_doc_cpf_cnpj`) VALUES
(1, 'Administrador',
'$2y$10$NAHCoamqGbr/JEx1l9nbpOpQ1D9DwM.lUXly3NViNMEOtDgPPrmZ2',
'ADMIN', 'admin@example.com', '999999999', 'Rua 1, 123', '123.456.789-01'),
(2, 'Usuário Teste',
'$2y$10$ouKD1dZXNfEMa2Qg/QcUMO1ONdL.rS9ZkzYkLmefwZnn4IopFKvV6',
'USUARIO', 'usuario@example.com', '988888888', 'Rua 2, 456', '987.654.321-00'),
(5, 'Andrei Admin',
'$2y$10$VSLKDlcFhFGVChLp6XiDYuSJNudRWcj90G0NySoTaFQhMZteIWWnm',
'ADMIN', 'teste@gmail.com', '(53) 99971-1852', 'Rua Testador Jafet Mil-anos, 3026',
'992.231.052-44'),
(6, 'usuario teste',
'$2y$10$USfTwu700aUcQ.ZS.vUT..pz6aZgMSbQ.zw5QjM4keyJZIWD9be1G',
'USUARIO', 'teste1@gmail.com', '6666666', 'rua ', '111.144.111-88'),
(7, 'otavio',
'$2y$10$OxrVLYgaLiaglzL1Tyxqke.QHqCSUmW3FDSsRYiXLkfBcK4YYuUvy',
'ADMIN', 'otavio@a.a', '(11) 11111-1111', 'ru22321', '222.222.222-22'),
(9, 'Coitado',
'$2y$10$xed0tfZGm7n9zUWMszaiBea4oTgpQLlsW9T9ZCo1pCPaQwUuqPrA.',
'USUARIO', 'coitado@1.com', '(53) 99921-1152', 'Rua General dos Coitados da Silva,
1903532', '17.121.161/1530-22'),
(10, 'aawodawoid',
'$2y$10$3JuVgEfwlJLE97DZ3QDLwOG1qzfLYSV6rbloTJe/Xx1VzPItBD15S',
'USUARIO', 'cu@gmail.com', '(12) 32312-1213', 'adwadawda', '32.121.313/2123-12');

-- -------------------------------------------------------

--
-- Estrutura para tabela `veiculo`
--
```

```
CREATE TABLE `veiculo` (
  `veiculo_id` int(11) NOT NULL,
  `veiculo_quilometragem` decimal(10,2) NOT NULL,
  `fk_Chassi_id` int(11) NOT NULL,
  `fk_Cor_id` int(11) NOT NULL,
  `fk_Modelo_id` int(11) NOT NULL,
  `fk_combustivel_id` int(11) NOT NULL,
  `veiculo_versao` varchar(60) NOT NULL,
  `veiculo_ano` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Despejando dados para a tabela `veiculo`
--

INSERT INTO `veiculo` (`veiculo_id`, `veiculo_quilometragem`, `fk_Chassi_id`,
`fk_Cor_id`, `fk_Modelo_id`, `fk_combustivel_id`, `veiculo_versao`, `veiculo_ano`)
VALUES
(9, 3400.00, 2, 2, 1, 1, 'Avant MHEV Diesel', 2021),
(25, 123000.00, 2, 4, 5, 4, '2.0 TITANIUM AWD 16V ', 2018),
(26, 10892.00, 1, 1, 7, 1, '4.0 Avant V8 Twinturbo MHEV Tiptronic', 2021),
(27, 139000.00, 2, 1, 10, 7, '2.4 16v Elite', 2006),
(28, 19900.00, 7, 4, 4, 1, '1.0 16v G3', 1999),
(29, 2120.00, 6, 9, 11, 1, '4.5L V8 Dual-Clutch', 2012),
(31, 1094.00, 6, 1, 12, 1, '6.2L Stingray V8 Targa', 2021),
(32, 0.00, 11, 1, 19, 1, '2.0 16V H4 Boxster PDK', 2026),
(33, 45900.00, 6, 2, 8, 1, '1.3 Manual', 1999),
(37, 21101.00, 7, 9, 25, 1, '3.0 V6 RS', 2004),
(38, 259000.00, 7, 2, 26, 1, '1.0 Hobby ', 1986);

--
-- Índices para tabelas despejadas
--

--
-- Índices de tabela `anuncio`
--
ALTER TABLE `anuncio`
  ADD PRIMARY KEY (`anuncio_id`),
  ADD KEY `fk_usuario_id` (`fk_usuario_id`),
  ADD KEY `fk_veiculo_id` (`fk_veiculo_id`);

--
```

```
-- Índices de tabela `chassi`
--
ALTER TABLE `chassi`
  ADD PRIMARY KEY (`chassi_id`);

--
-- Índices de tabela `combustivel`
--
ALTER TABLE `combustivel`
  ADD PRIMARY KEY (`comb_id`);

--
-- Índices de tabela `cor`
--
ALTER TABLE `cor`
  ADD PRIMARY KEY (`cor_id`);

--
-- Índices de tabela `imagem`
--
ALTER TABLE `imagem`
  ADD PRIMARY KEY (`imagem_id`),
  ADD UNIQUE KEY `imagem_url` (`imagem_url`),
  ADD KEY `fk_anuncio_id` (`fk_anuncio_id`);

--
-- Índices de tabela `marca`
--
ALTER TABLE `marca`
  ADD PRIMARY KEY (`marca_id`);

--
-- Índices de tabela `modelo`
--
ALTER TABLE `modelo`
  ADD PRIMARY KEY (`modelo_id`),
  ADD KEY `fk_Marca_id` (`fk_Marca_id`);

--
-- Índices de tabela `usuario`
--
ALTER TABLE `usuario`
  ADD PRIMARY KEY (`usuario_id`);
```

```
--
-- Índices de tabela `veiculo`
--
ALTER TABLE `veiculo`
  ADD PRIMARY KEY (`veiculo_id`),
  ADD KEY `fk_Chassi_id` (`fk_Chassi_id`),
  ADD KEY `fk_Cor_id` (`fk_Cor_id`),
  ADD KEY `fk_Modelo_id` (`fk_Modelo_id`),
  ADD KEY `fk_combustivel_id` (`fk_combustivel_id`);

--
-- AUTO_INCREMENT para tabelas despejadas
--

--
-- AUTO_INCREMENT de tabela `anuncio`
--
ALTER TABLE `anuncio`
  MODIFY `anuncio_id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=38;

--
-- AUTO_INCREMENT de tabela `chassi`
--
ALTER TABLE `chassi`
  MODIFY `chassi_id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=14;

--
-- AUTO_INCREMENT de tabela `combustivel`
--
ALTER TABLE `combustivel`
  MODIFY `comb_id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=8;

--
-- AUTO_INCREMENT de tabela `cor`
--
ALTER TABLE `cor`
  MODIFY `cor_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=17;

--
-- AUTO_INCREMENT de tabela `imagem`
--
```

```
ALTER TABLE `imagem`
  MODIFY `imagem_id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=97;

--
-- AUTO_INCREMENT de tabela `marca`
--
ALTER TABLE `marca`
  MODIFY `marca_id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=23;

--
-- AUTO_INCREMENT de tabela `modelo`
--
ALTER TABLE `modelo`
  MODIFY `modelo_id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=27;

--
-- AUTO_INCREMENT de tabela `usuario`
--
ALTER TABLE `usuario`
  MODIFY `usuario_id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=21;

--
-- AUTO_INCREMENT de tabela `veiculo`
--
ALTER TABLE `veiculo`
  MODIFY `veiculo_id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=40;

--
-- Restrições para tabelas despejadas
--

--
-- Restrições para tabelas `anuncio`
--
ALTER TABLE `anuncio`
  ADD CONSTRAINT `anuncio_ibfk_1` FOREIGN KEY (`fk_usuario_id`)
REFERENCES `usuario` (`usuario_id`),
  ADD CONSTRAINT `anuncio_ibfk_2` FOREIGN KEY (`fk_veiculo_id`)
REFERENCES `veiculo` (`veiculo_id`);
```

```
--
-- Restrições para tabelas `imagem`
--
ALTER TABLE `imagem`
  ADD CONSTRAINT `imagem_ibfk_1` FOREIGN KEY (`fk_anuncio_id`)
REFERENCES `anuncio` (`anuncio_id`) ON DELETE CASCADE;

--
-- Restrições para tabelas `modelo`
--
ALTER TABLE `modelo`
  ADD CONSTRAINT `modelo_ibfk_1` FOREIGN KEY (`fk_Marca_id`)
REFERENCES `marca` (`marca_id`);

--
-- Restrições para tabelas `veiculo`
--
ALTER TABLE `veiculo`
  ADD CONSTRAINT `veiculo_ibfk_1` FOREIGN KEY (`fk_Chassi_id`)
REFERENCES `chassi` (`chassi_id`),
  ADD CONSTRAINT `veiculo_ibfk_2` FOREIGN KEY (`fk_Cor_id`) REFERENCES
`cor` (`cor_id`),
  ADD CONSTRAINT `veiculo_ibfk_3` FOREIGN KEY (`fk_Modelo_id`)
REFERENCES `modelo` (`modelo_id`),
  ADD CONSTRAINT `veiculo_ibfk_4` FOREIGN KEY (`fk_combustivel_id`)
REFERENCES `combustivel` (`comb_id`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```