
Documentação de Projeto

para o sistema

BusCars

Versão 1.0

Projeto de sistema elaborado pelo(s) aluno(s) Lucas Araujo Borges de Lima e Luis Gustavo Vaz e apresentado ao curso de **Engenharia de Software** da **PUC Minas** como parte do Trabalho de Conclusão de Curso (TCC) sob orientação de conteúdo do professor João Paulo Aramuni, orientação acadêmica do professor Cleiton Silva Tavares e orientação de TCC II do professor João Paulo Aramuni.

08/08/2025

Tabela de Conteúdo

1. Introdução	1
2. Modelos de Usuário e Requisitos	1
2.1 Descrição de Atores	2
2.2 Modelos de Usuários	2
2.3 Modelo de Casos de Uso e Histórias de Usuários	4
2.3.1 Diagrama de Caso de Uso	4
2.3.2 Historias de Usuário	6
2.4 Diagrama de Sequência do Sistema e Contrato de Operações	7
3. Modelos de Projeto	7
3.1 Diagrama de Classes	7
3.2 Diagramas de Sequência	7
3.3 Diagramas de Comunicação	7
3.4 Arquitetura	7
3.4.1 Motivação da stack e das ferramentas	7
3.4.2 Descrição da arquitetura	8
3.5 Diagramas de Estados	10
3.6 Diagrama de Componentes e Implantação.	10
4. Projeto de Interface com Usuário	10
4.1 Esboço das Interfaces Comuns a Todos os Atores	10
4.2 Esboço das Interfaces Usadas pelo Ator <A>	14
4.3 Esboço das Interfaces Usadas pelo Ator 	14
5. Glossário e Modelos de Dados	15
6. Casos de Teste	15
7. Cronograma e Processo de Implementação	15

Histórico de Revisões

Nome	Data	Razões para Mudança	Versão
Entrega 1	07/09/25	Finalização do documento do projeto BusCars	1.0

1. Introdução

Este documento de projeto do sistema **BusCars** serve como um alicerce técnico e estrutural para a elaboração e implementação de uma plataforma robusta e intuitiva de agregação e gerenciamento de veículos. A intenção primária deste documento é proporcionar uma visão detalhada dos modelos de domínio que fundamentam o sistema, assim como dos modelos de design que orientam sua construção. O documento é essencialmente técnico e destina-se a guiar a equipe de desenvolvimento via uma série de especificações meticulosamente organizadas, desenhos de arquitetura, e diretrizes de implementação em sincronia com os requisitos e objetivos delineados no documento de visão do sistema.

A funcionalidade e a inovação do sistema BusCars residem na sua capacidade de transformar e modernizar a maneira como a pesquisa e a compra de veículos são realizadas no mercado automotivo brasileiro. O sistema BusCars aborda como a plataforma irá melhorar a eficiência da busca, padronizar as informações de diferentes fontes e aprimorar a experiência do usuário por meio de uma interface amigável e de fácil utilização.

Por meio desta documentação, busca-se definir os padrões técnicos e os procedimentos necessários para garantir uma integração e escalabilidade eficazes, contemplando a aplicação de tecnologias avançadas para otimizar as operações e fortalecer a segurança do sistema. Este projeto foi desenvolvido para atender diretamente às necessidades e expectativas de **Lucas Amaral Paes Leme Maestro**, o stakeholder e gestor no setor automotivo, que deseja digitalizar e expandir a operação de compra e venda de veículos.

Lucas Amaral tem uma experiência significativa no setor automotivo na região metropolitana de Belo Horizonte. A digitalização da operação por meio da aplicação BusCars permitirá a otimização de seu fluxo de trabalho, que hoje exige a consulta manual de diversas plataformas, para uma gestão centralizada e mais eficiente. Com a implementação do sistema, o objetivo é tornar a curadoria dos anúncios de veículos mais eficazes, abrangendo desde a busca de dados em diferentes bases até a visualização e comparação, proporcionando uma experiência completa para os usuários finais.

Assim, a aplicação BusCars foi idealizada não apenas para modernizar a gestão interna do serviço, mas também para impulsionar o crescimento do negócio, proporcionando um sistema confiável, ágil e de fácil acesso tanto para compradores quanto para vendedores.

2. Modelos de Usuário e Requisitos

Nesta seção serão apresentadas as informações sobre os usuários e os requisitos do projeto. Além disso, serão apresentados o diagrama de caso de uso e as histórias de usuário.

2.1 Descrição de Atores

Os atores identificados para a utilização do sistema BusCars são os Clientes, os Vendedores, o Stakeholder e os Administradores. Os **Clientes** são os usuários finais que buscam veículos, utilizando a plataforma para centralizar suas pesquisas e obter informações detalhadas sobre as ofertas do mercado. **Vendedores** são os profissionais do setor que utilizam o sistema para acompanhar o mercado e, no futuro, poderão publicar seus próprios anúncios com maior riqueza de detalhes. O **Stakeholder** é o visionário do projeto, responsável por fornecer o conhecimento do mercado automotivo e garantir que o produto atenda às necessidades do setor. Por fim, os **Administradores** são os responsáveis pela gestão e operação do sistema, cuidando da integridade dos dados, monitorando as coletas e garantindo a eficiência da plataforma. Todos esses atores colaboram para que o BusCars seja uma solução completa e confiável no cenário de compra e venda de veículos.

2.2 Modelos de Usuários

Esta seção tem o propósito de descrever os modelos de usuários desenvolvidos por meio da implementação de personas. Com esse intuito, foram elaboradas personas representativas que funcionarão como orientações durante o processo de design e desenvolvimento da plataforma. Essas personas foram construídas com base em uma compreensão aprofundada das necessidades e características dos usuários-alvo. Na Tabela 1 é apresentada a persona do **Cliente**, representando o usuário que busca e compara veículos na plataforma. Na Tabela 2 é apresentada a persona do **Vendedor**, representando o profissional que acompanha o mercado e, futuramente, poderá publicar seus próprios anúncios. Já na Tabela 3 é apresentada a persona do **Administrador**, representando o usuário responsável por gerenciar e coordenar a base de dados do sistema.

Eduardo Pena	
Descrição	Ana Beatriz é uma profissional ocupada que vive em Belo Horizonte e está em busca de um veículo seminovo para uso diário. Ela valoriza seu tempo e busca uma forma mais eficiente e confiável de pesquisar carros, evitando ter que visitar múltiplos sites e lojas físicas sem uma pré-seleção. Ela é bastante conectada e utiliza a internet como sua principal ferramenta de pesquisa.
Dores	<ul style="list-style-type: none">• A busca por veículos é fragmentada, exigindo que ela acesse vários marketplaces diferentes, o que é demorado e confuso.• A falta de padronização nas informações dificulta a comparação direta entre os anúncios.• Ela se sente insegura sobre a real condição e o preço justo de um veículo.
Objetivos	<ul style="list-style-type: none">• Encontrar o carro ideal que se encaixe em seu orçamento e necessidades.• Tomar uma decisão de compra mais informada e segura, com acesso a dados comparativos.

	<ul style="list-style-type: none"> • Economizar tempo no processo de pesquisa, centralizando a busca em uma única plataforma confiável.
Tarefas	<ul style="list-style-type: none"> • Utilizar a barra de busca e os filtros avançados da plataforma. • Comparar diferentes ofertas de veículos. • Visualizar o preço de referência da Tabela FIPE para um anúncio. • Identificar anúncios duplicados para evitar perder tempo com ofertas repetidas. • Salvar buscas e veículos favoritos para revisão posterior.

Tabela 1. Persona Eduardo Pena

Henrique Mota	
Descrição	Henrique Mota é proprietário de uma pequena revenda de carros seminovos em Belo Horizonte. Ele tem um estoque limitado e, por isso, busca otimizar a visibilidade de seus veículos para atrair clientes qualificados. Atualmente, ele utiliza vários marketplaces, mas percebe que as plataformas existentes não valorizam o diferencial de seu estoque.
Dores	<ul style="list-style-type: none"> • Falta de visibilidade e destaque para seus veículos em meio a um grande volume de anúncios genéricos. • O gerenciamento de anúncios em diferentes plataformas é demorado e ineficiente. • A dificuldade em atrair leads qualificados e em lidar com contatos de baixo valor ou curiosos, que não resultam em vendas concretas.
Objetivos	<ul style="list-style-type: none"> • Aumentar a visibilidade de seu estoque para atrair clientes mais qualificados. • Simplificar o gerenciamento de seus anúncios em um único lugar. • Utilizar uma plataforma que valorize a qualidade de seus veículos e permita a criação de anúncios mais completos e atrativos.
Tarefas	<ul style="list-style-type: none"> • Acompanhar os preços de mercado na plataforma para precificar seu estoque de forma competitiva. • Buscar veículos para compra em revendas. • No futuro, criar e gerenciar anúncios próprios, adicionando descrições detalhadas e fotos de alta resolução.

Tabela 2. Persona Henrique Mota

Caio Silva	
Descrição	Caio Silva é o responsável pela gestão e operação da plataforma BusCars. Ele atua nos bastidores, garantindo que o sistema funcione de forma fluida, confiável e segura. Sua principal função é monitorar a integridade das informações e manter a saúde técnica do produto. Ele garante a qualidade e consistência do BusCars, assegurando que a experiência do usuário seja fluida apesar da complexa integração de dados.
Dores	<ul style="list-style-type: none">● Falta de visibilidade sobre falhas de integração (web scraping e APIs).● Inconsistência nos dados agregados, que podem não estar padronizados ou desatualizados.● A identificação manual de anúncios duplicados ou fraudulentos consome muito tempo e recursos.
Objetivos	<ul style="list-style-type: none">● Garantir a operação contínua e a alta disponibilidade do sistema.● Assegurar a qualidade e a consistência dos dados apresentados aos usuários.● Otimizar o processo de moderação e manutenção da base de dados
Tarefas	<ul style="list-style-type: none">● Monitorar o log de erros das integrações com outras plataformas.● Validar novos dados e remover anúncios inconsistentes ou duplicados.● Acompanhar métricas de desempenho do sistema e do banco de dados.● Realizar atualizações e manutenções na plataforma.

Tabela 3. Persona Caio Silva

2.3 Modelo de Casos de Uso e Histórias de Usuários

Nesta subseção são apresentados o diagrama de casos de uso do sistema e as histórias dos usuários, que serão apresentadas nas seções 2.3.1 e 2.3.2 respectivamente.

2.3.1 Diagrama de Caso de Uso

A Figura 1 representa o diagrama de casos de uso referentes ao sistema proposto. No diagrama é possível ver 3 atores principais, Cliente, Vendedor, e Administrador, que são usuários do sistema.

A busca é a principal funcionalidade do BusCars, o ponto de entrada para a experiência de todos os usuários, sejam eles Clientes ou Vendedores. Diferente dos marketplaces tradicionais, que apresentam dados de forma isolada, o BusCars consolida informações de diversas fontes para oferecer uma pesquisa centralizada e inteligente.

Para o Cliente, a busca é a ferramenta fundamental para encontrar o carro ideal. A plataforma permite que ele pesquise utilizando filtros avançados, como marca, modelo, ano e localização, e acesse em uma única tela os anúncios unificados de diferentes sites. Isso não apenas poupa tempo, mas também oferece uma visão comparativa do mercado, com dados enriquecidos como o valor da Tabela FIPE.

Já para o Vendedor, a busca serve como uma poderosa ferramenta de inteligência de mercado. Ao utilizar os mesmos filtros, ele pode analisar o comportamento dos preços de mercado para precificar seu próprio estoque de forma mais competitiva. Além disso, a busca por oportunidades de compra permite que ele encontre veículos com potencial de revenda, otimizando seu negócio.

O Administrador opera nos bastidores do BusCars garantindo a integridade e a saúde da plataforma. Sua função principal é monitorar a coleta de dados para assegurar que os anúncios sejam importados sem falhas. Ele também valida o conteúdo, removendo informações inconsistentes ou fraudulentas, e identifica duplicatas para manter a base de dados limpa. Para tomar decisões estratégicas, ele utiliza a visualização de métricas do sistema, analisando o desempenho e o comportamento dos usuários.

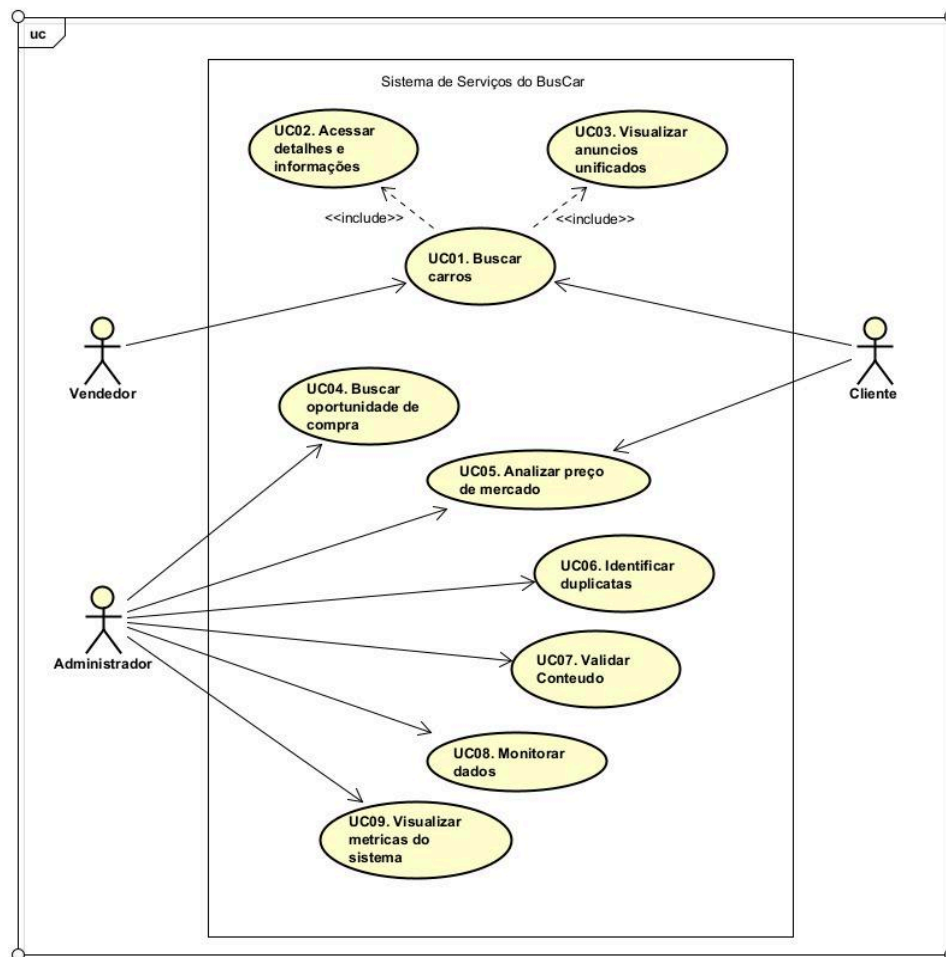


Figura 1. Diagrama de Caso de Uso

2.3.2 Histórias de Usuário

Nesta seção são listadas as histórias de usuários elicitadas para o sistema do Chaveiro Mestre. Para fins de organização, utilizam-se identificadores no formato US#ID, em que US refere-se a *User Story*. Assim, as histórias de usuário identificadas para o sistema são:

US01. Como Vendedor, desejo buscar carros na plataforma, para encontrar veículos com potencial de revenda e analisar o mercado.

US02. Como Vendedor, desejo poder acessar os detalhes e as informações de um veículo, para ter os dados completos e tomar uma decisão de compra mais informada.

US03. Como Vendedor, desejo visualizar todos os anúncios de um veículo em uma única página, para comparar as ofertas de forma rápida e eficiente.

US04. Como Cliente, desejo buscar veículos na plataforma, para encontrar o carro ideal que se encaixe nas minhas necessidades.

US05. Como Cliente, desejo poder acessar os detalhes e as informações de um veículo, para ter uma visão completa antes de tomar uma decisão.

US06. Como Cliente, desejo visualizar os anúncios de um veículo em uma única página, para evitar ter que abrir várias abas no navegador e poupar tempo.

US07. Como Cliente, desejo analisar o preço de um carro em relação ao mercado, para saber se a oferta é justa.

US08. Como Administrador, preciso analisar o preço médio de um veículo, para manter a base de dados da plataforma com informações precisas.

US09. Como Administrador, preciso identificar anúncios duplicados na base de dados, para garantir a qualidade e a unicidade das informações.

US10. Como Administrador, preciso validar o conteúdo dos anúncios, para remover dados inconsistentes e garantir a credibilidade da plataforma.

US11. Como Administrador, desejo monitorar a coleta de dados de cada fonte, para garantir que o sistema esteja sempre atualizado.

US12. Como Administrador, desejo visualizar as métricas de desempenho do sistema, para entender o comportamento dos usuários e guiar futuras melhorias.

2.4 Diagrama de Sequência do Sistema e Contrato de Operações

Nesta subseção é apresentado o diagrama de sequência do sistema e os Contratos de Operações.

Formato para cada contrato de operação

Contrato	
Operação	
Referências cruzadas	
Pré-condições	
Pós-condições	

3. Modelos de Projeto

3.1 Diagrama de Classes

Diagrama de classes do sistema

3.2 Diagramas de Sequência

Diagramas de sequência para realização de casos de uso.

3.3 Diagramas de Comunicação

Diagramas de comunicação para realização de casos de uso.

3.4 Arquitetura

3.4.1 Motivação da stack e das ferramentas

A escolha da stack tecnológica do sistema BusCars fundamenta-se na necessidade de equilibrar desempenho, confiabilidade, segurança e facilidade de manutenção ao longo de todo o ciclo de vida do projeto. Considerando-se que se trata de uma plataforma web voltada à agregação de anúncios automotivos provenientes de diferentes fontes externas, tornou-se indispensável selecionar ferramentas capazes de lidar com grandes volumes de dados, prover mecanismos de cache e escalabilidade, além de garantir que o produto final seja acessível, estável e de fácil implantação em diferentes ambientes.

A linguagem OCaml, associada ao framework Dream, foi selecionada como base do desenvolvimento por possibilitar alto grau de segurança em tempo de compilação, expressividade e concisão no código-fonte. A tipagem forte e estática oferecida pelo OCaml contribui para a redução de erros em tempo de execução e promove maior robustez da aplicação, característica essencial em um sistema que realiza integração com múltiplos marketplaces. O framework Dream, por sua vez, viabiliza a construção de aplicações web de forma direta e moderna, com suporte a renderização de HTML e integração nativa a componentes de autenticação, roteamento e manipulação de requisições HTTP. A adoção dessa combinação também favorece a consistência técnica do projeto e a experimentação de paradigmas funcionais que simplificam a lógica de negócio.

O PostgreSQL foi escolhido como sistema gerenciador de banco de dados por se tratar de uma solução madura, estável e amplamente reconhecida na indústria. Entre seus principais diferenciais estão o suporte a dados semi estruturados (como JSONB), mecanismos avançados de indexação (GIN e GiST) e extensões para geolocalização, recursos que se alinham às demandas de busca, filtragem e análise comparativa dos anúncios automotivos. Essas funcionalidades permitem ao sistema realizar consultas complexas de forma eficiente e fornecer indicadores enriquecidos, como valores médios de mercado e referências à Tabela FIPE.

Para lidar com operações de alta frequência, sessões de usuários e armazenamento temporário de resultados, foi adotado o Redis. A ferramenta desempenha um papel estratégico como mecanismo de cache e também como sistema de filas, permitindo processar integrações externas e normalização de dados de forma assíncrona, sem comprometer a responsividade da aplicação.

A utilização de Docker e Docker Compose justifica-se pela necessidade de padronizar os ambientes de desenvolvimento, homologação e produção. Essa abordagem assegura que todos os serviços (frontend, backend, banco de dados, cache e proxy) sejam executados de maneira consistente e portátil, reduzindo riscos de incompatibilidade entre diferentes sistemas operacionais ou configurações de servidores. Além disso, possibilita a rápida replicação do ambiente por qualquer membro da equipe, o que contribui para a agilidade do processo de desenvolvimento.

O Nginx foi incorporado à arquitetura como servidor web e proxy reverso, responsável pelo roteamento interno entre os contêineres, pela compressão de tráfego e pela otimização de conexões HTTPS. Em conjunto, a plataforma conta ainda com os recursos do Cloudflare, que atua como camada adicional de distribuição de conteúdo e segurança. A integração com a rede global da Cloudflare garante baixa latência, disponibilidade elevada e proteção contra ataques de negação de serviço (DDoS), oferecendo maior resiliência e confiabilidade ao sistema.

3.4.2 Descrição da arquitetura

De forma resumida, a arquitetura do BusCars pode ser representada conforme o diagrama abaixo:

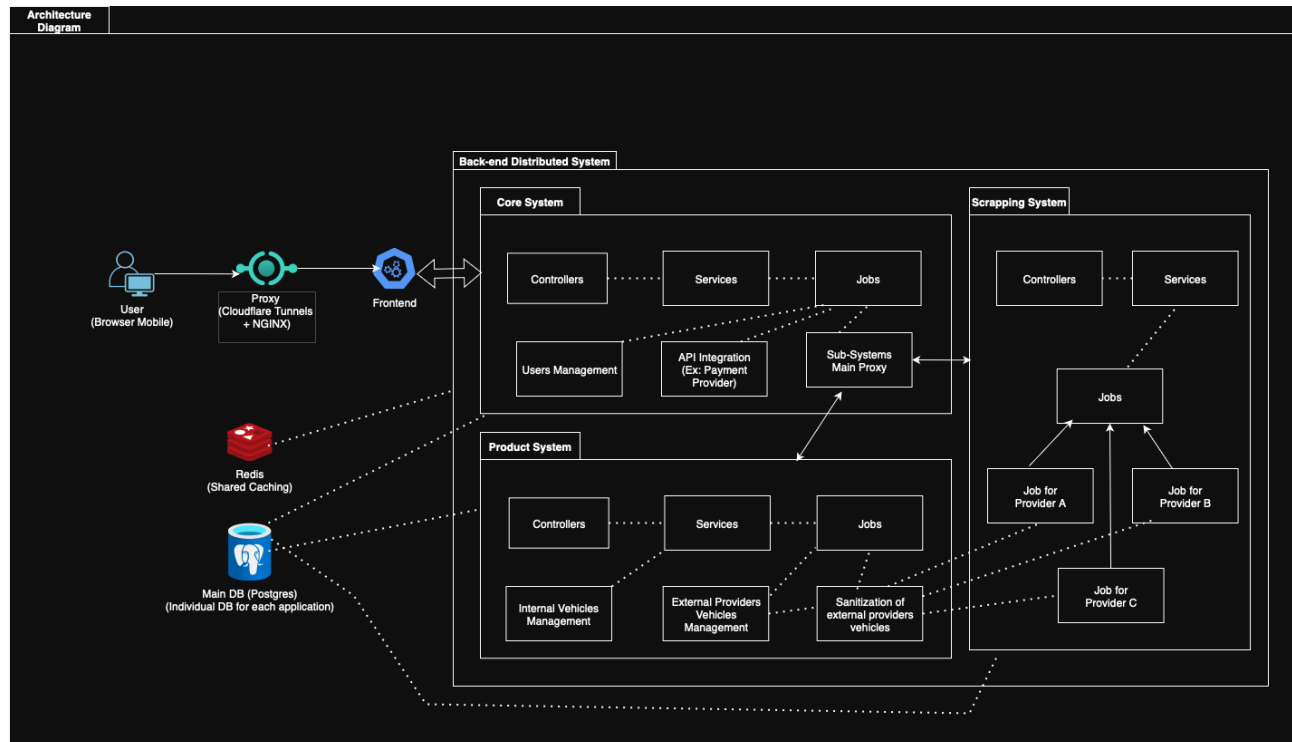


Figura 2. Diagrama de Arquitetura

A Figura 2 apresenta o diagrama de arquitetura, detalhando o sistema BusCars, a arquitetura foi concebida em camadas, de modo a assegurar a separação clara de responsabilidades, a manutenibilidade do código e a escalabilidade da solução. No nível superior, encontra-se a interface web, responsável por oferecer aos usuários finais uma experiência responsiva e intuitiva de busca e comparação de veículos. Essa interface é construída em HTML e CSS e entregue pelo framework Dream, que, além de gerenciar a camada de apresentação, também integra a lógica de controle das requisições.

No núcleo do sistema está o backend, igualmente desenvolvido em OCaml com o framework Dream. Ele desempenha o papel de camada de aplicação, recebendo requisições da interface, processando regras de negócio e interagindo com os serviços de persistência e cache. É nesta camada que se encontram as funcionalidades de agregação, normalização de dados, deduplicação de anúncios e cálculo de indicadores de mercado, assegurando que as informações apresentadas ao usuário sejam consistentes e enriquecidas.

O PostgreSQL constitui a camada de persistência, armazenando de forma estruturada todos os dados do sistema, incluindo usuários, anúncios importados de diferentes marketplaces, logs de integrações e métricas derivadas. O modelo de dados é projetado de forma a contemplar tanto atributos comuns (marca, modelo, ano, quilometragem, preço) quanto campos adicionais relacionados a padronização e enriquecimento das informações, permitindo consultas complexas e relatórios consolidados.

Complementarmente, o Redis exerce a função de cache e mecanismo de filas, contribuindo para a otimização da performance. Requisições de busca recorrentes podem ser rapidamente respondidas a partir do cache, enquanto filas são utilizadas para coordenar processos assíncronos, como a atualização de anúncios via APIs ou rotinas de web scraping. Essa abordagem evita sobrecarga no banco de dados principal e melhora a experiência do usuário, que recebe respostas mais rápidas.

Na camada de infraestrutura, o Nginx atua como proxy reverso, recebendo as conexões externas e direcionando-as adequadamente para os serviços internos executados em contêineres. Essa camada também

implementa recursos de balanceamento de carga e compressão, além de interagir diretamente com os serviços de CDN e proteção fornecidos pela Cloudflare. O Cloudflare, por sua vez, garante resiliência, otimização de entrega de conteúdo e segurança contra ameaças externas, compondo a linha de frente da arquitetura.

Por fim, toda a solução é orquestrada por meio do Docker Compose, que descreve a configuração dos diferentes serviços, suas dependências e a rede interna de comunicação. Essa abordagem facilita a implantação tanto em servidores locais quanto em ambientes de nuvem, permitindo que a equipe mantenha controle sobre a escalabilidade e o monitoramento de cada componente.

3.5 Diagramas de Estados

Diagramas de estados do sistema.

3.6 Diagrama de Componentes e Implantação.

Diagramas de componentes do sistema. Diagrama de implantação mostrando onde os componentes estarão alocados para a execução.

4. Projeto de Interface com Usuário

Esta seção tem como objetivo demonstrar e descrever as interfaces de interação com o usuário que compõem a aplicação. Para isso, foram utilizados mockups de um projeto demo. As interfaces foram correlacionadas aos casos de uso especificados na Seção 2.3.1, mapeando todas as funcionalidades necessárias para o atendimento dos requisitos estabelecidos.

4.1 Esboço das Interfaces Comuns a Todos os Atores



Figura 3. Tela inicial

A tela inicial, representada na Figura 3, do BusCar atua como o ponto de partida central para o usuário. No topo, um **menu de navegação** garante acesso rápido a outras seções da plataforma, como a área de login para usuários já cadastrados. O destaque da página, no entanto, é o **buscador principal de veículos**. Ele é posicionado de forma proeminente para convidar o usuário a iniciar sua pesquisa imediatamente, utilizando campos intuitivos para refinar a busca por marca, modelo ou ano.

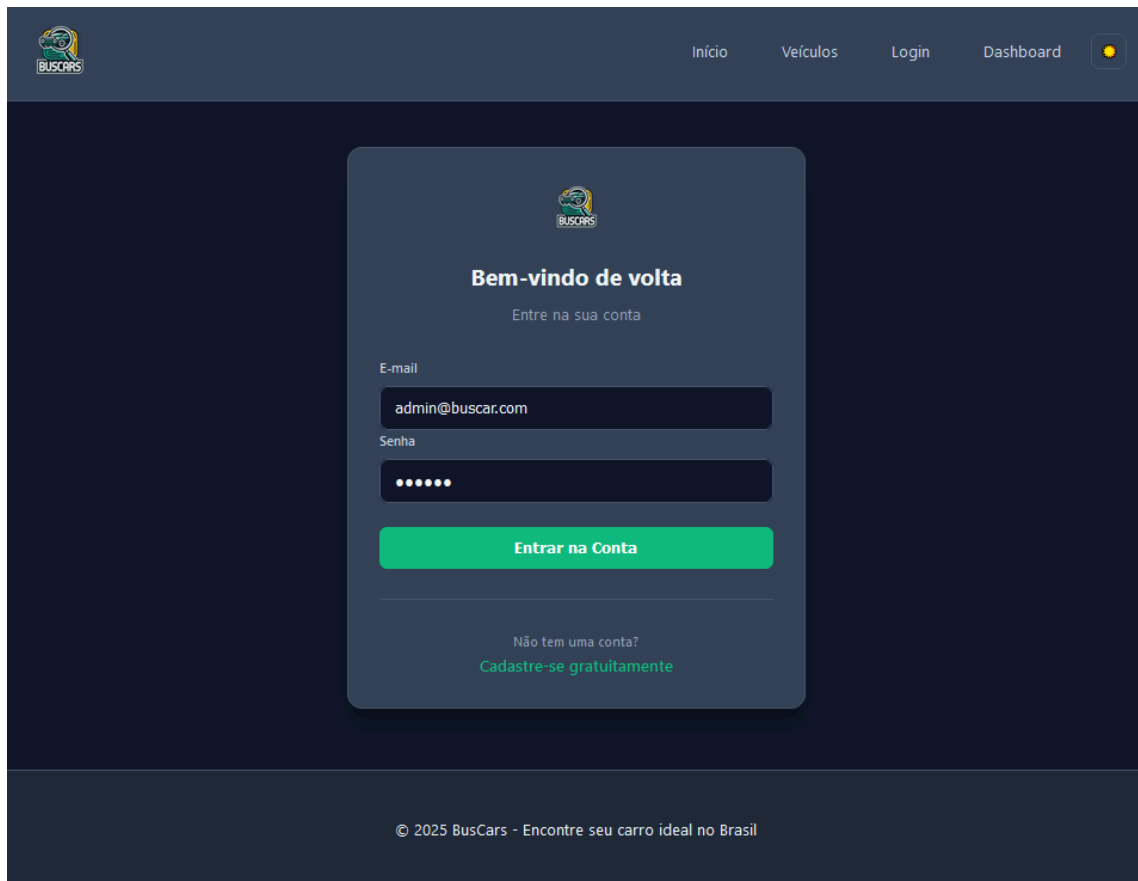


Figura 4. Tela de Login

A tela inicial, representada na Figura 4, serve como o ponto de entrada principal para a plataforma. No topo, ela oferece campos para **login**, permitindo que usuários já registrados acessem suas contas. No centro da tela, um campo de busca proeminente convida tanto o usuário logado quanto o visitante a iniciar uma pesquisa por veículos. O design é limpo e direto, priorizando a usabilidade para que a pesquisa comece imediatamente.

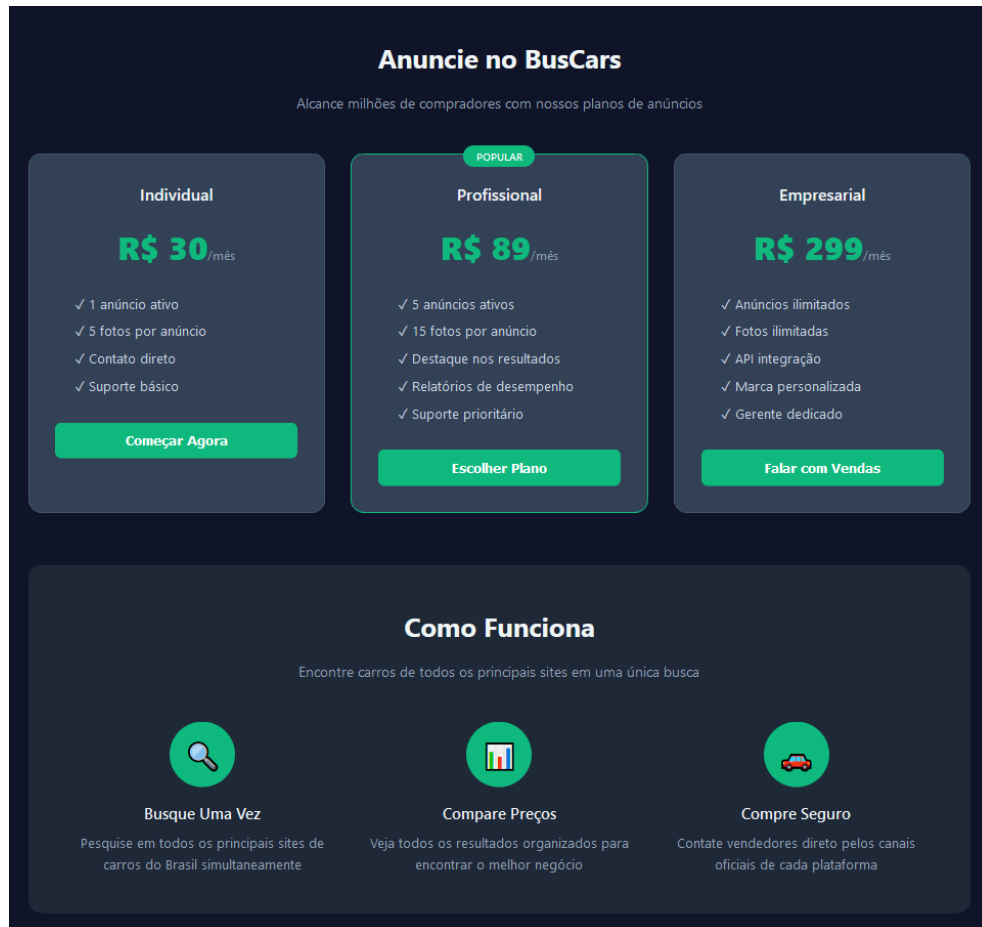


Figura 5. Tela de informações e planos

A tela de informações e planos, representada na Figura 5, do BusCars foi criada para educar os usuários sobre o uso da plataforma e apresentar as opções de monetização. Ela detalha o funcionamento da ferramenta de busca e dos filtros para os compradores, enquanto para os vendedores, descreve os planos de assinatura disponíveis, com seus respectivos benefícios e custos, como a quantidade de anúncios e o acesso a ferramentas de análise. O objetivo principal é garantir total transparência sobre o modelo de negócio e as funcionalidades oferecidas em cada nível de serviço.

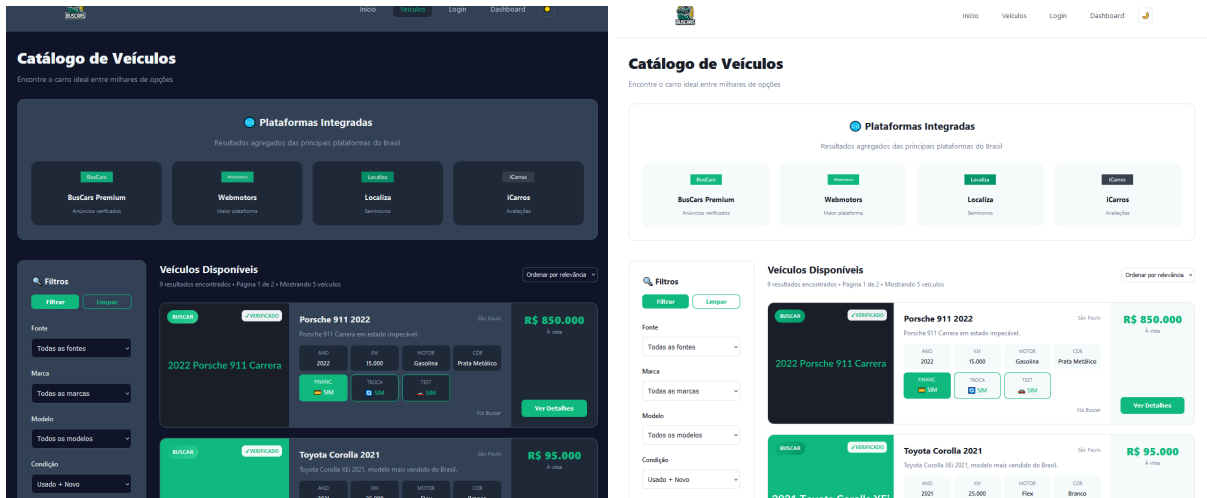


Figura 6. Tela de veículos

Esta tela de veículos, representada na Figura 6, é o coração da experiência de busca. Nela, os resultados da pesquisa são exibidos em um formato de grade ou lista. Cada cartão de anúncio apresenta as informações-chave de forma unificada e padronizada, independentemente da fonte original (como WebMotors ou Icarros). Essa **visualização unificada** é o principal diferencial do BusCars, facilitando a comparação visual entre diferentes veículos.

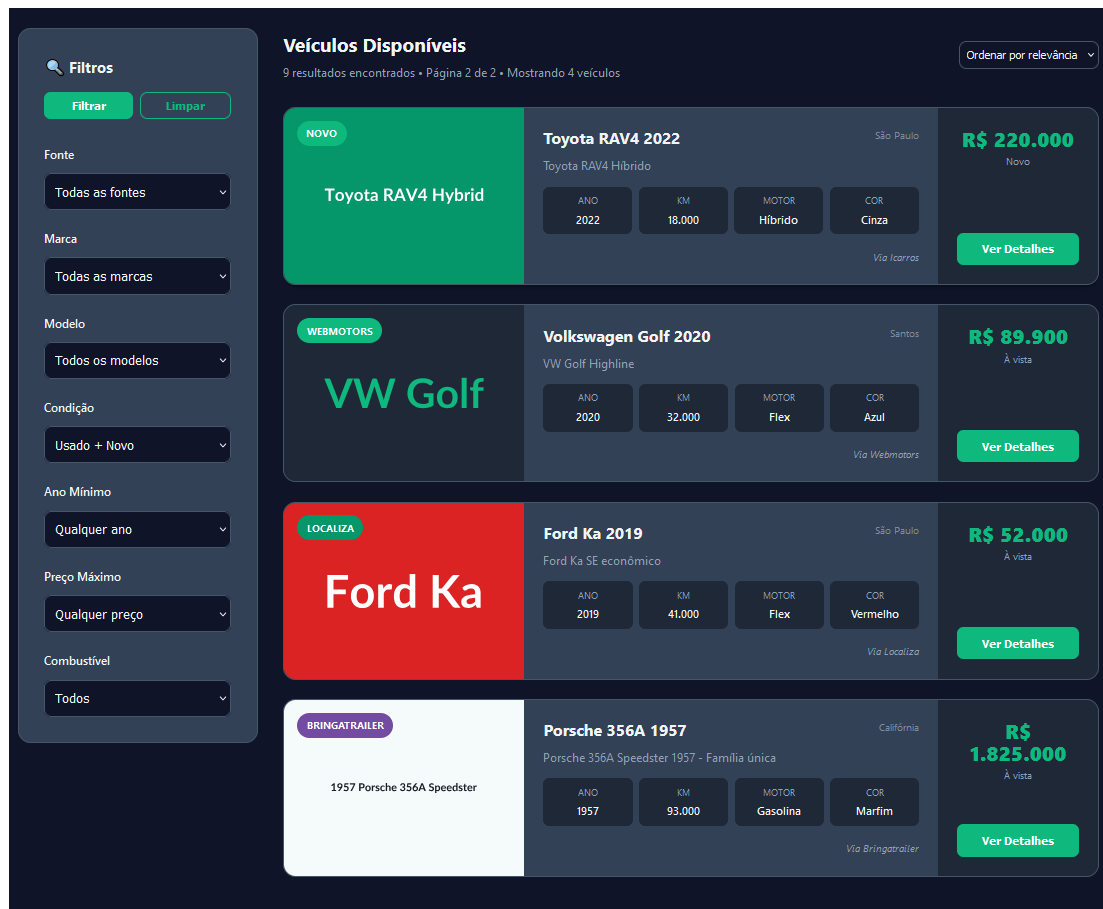


Figura 7. Tela de filtro de veículos

Após iniciar a busca, o usuário é direcionado para a tela de resultados, representada na Figura 7. Nela, a seção de **filtros de busca** se destaca como a ferramenta essencial para refinar a pesquisa. O usuário pode aplicar diversos critérios, como **marca, modelo, ano, faixa de preço e quilometragem**, para personalizar os resultados e encontrar exatamente o que procura, eliminando anúncios irrelevantes.

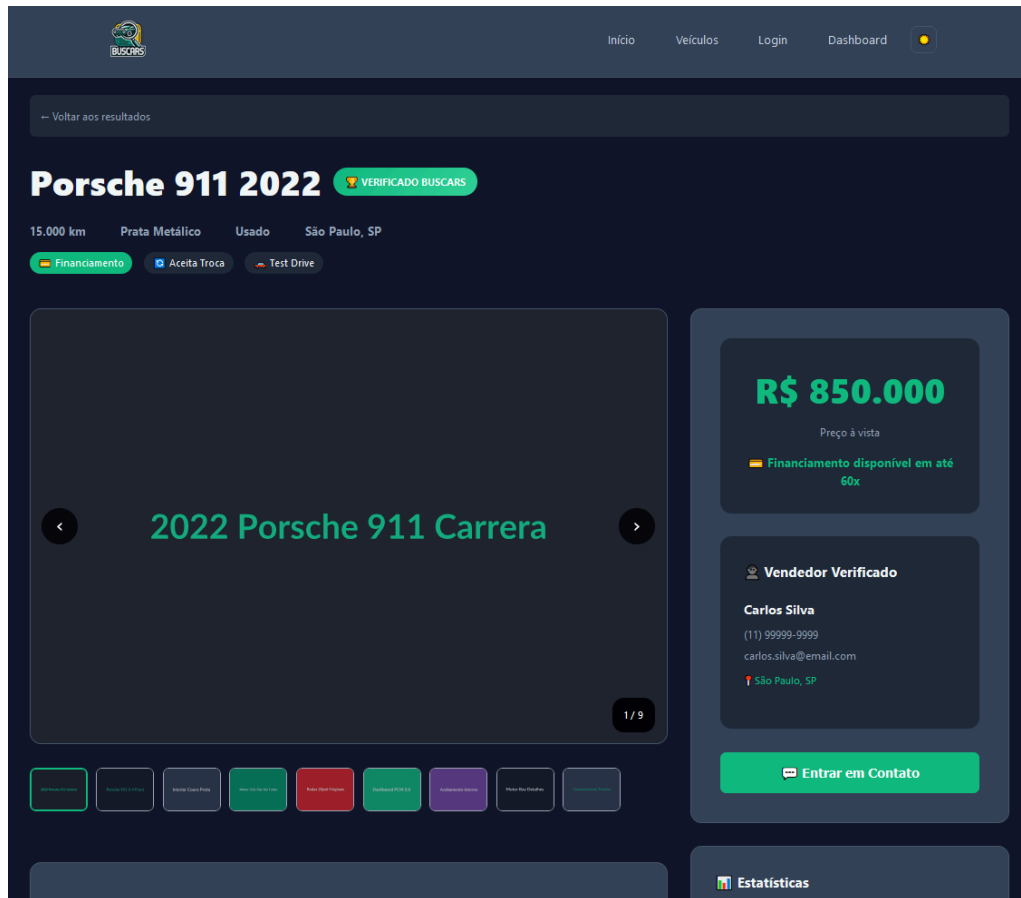


Figura 8. Tela de informações do veículo

Ao clicar em um veículo na tela de visualização, representada na Figura 8, o usuário acessa a página de **detalhes completos**. Aqui, todas as informações do anúncio são apresentadas de forma aprofundada. Além dos dados básicos como fotos, descrição e contato do vendedor, o sistema enriquece o conteúdo com **informações adicionais**, como o valor de referência da **Tabela FIPE**, ajudando o comprador a ter uma visão completa e segura sobre o valor do veículo.

4.2 Esboço das Interfaces Usadas pelo Ator <A>

Wireframe/mockup/storyboard das interfaces exclusivas do ator <A>

4.3 Esboço das Interfaces Usadas pelo Ator

Wireframe/mockup/storyboard das interfaces exclusivas do ator

5. Glossário e Modelos de Dados

Deve-se apresentar o glossário para o sistema. Também apresente esquemas de banco de dados e as estratégias de mapeamento entre as representações de objetos e não-objetos.

6. Casos de Teste

Uma descrição de casos de teste para validação do sistema.

7. Cronograma e Processo de Implementação

O desenvolvimento do projeto está programado para ocorrer ao longo de quatro meses e meio, com início em agosto de 2025 e conclusão prevista para a segunda quinzena de dezembro de 2025. As atividades de desenvolvimento foram organizadas em issues e divididas em milestones, gerenciadas por meio do GitHub do projeto. O processo de desenvolvimento será estruturado em sprints, cada um com aproximadamente 15 dias de duração. Ao final de cada iteração, haverá uma reunião de alinhamento com o stakeholder para fornecer feedback e realizar eventuais ajustes no escopo do projeto.

Milestones	Luis Gustavo	Lucas Lima
1ª Quinzena de Agosto	<ul style="list-style-type: none">• Criação e organização dos boards no GitHub para o gerenciamento das tarefas.• Provisionamento e configuração da VPS (Virtual Private Server) para o ambiente de testes.• Criação do repositório Git e configuração de branches.	<ul style="list-style-type: none">• Configuração do ambiente de desenvolvimento (IDE, Docker).• Definição da arquitetura inicial do sistema.
2ª Quinzena de Agosto	<ul style="list-style-type: none">• Revisão e definição do modelo de dados para a agregação de veículos.• Criação das tabelas no banco de dados (PostgreSQL) para armazenar os dados dos anúncios.	<ul style="list-style-type: none">• Implementação da estrutura inicial do projeto (backend e frontend).• Documentação inicial do ambiente configurado.
1ª Quinzena de Setembro	<ul style="list-style-type: none">• Desenvolvimento de rotinas de web scraping e/ou integração	<ul style="list-style-type: none">• Criação de endpoints de API para receber e processar os dados dos

	com APIs para a coleta de anúncios de diferentes marketplaces.	agregadores. <ul style="list-style-type: none"> Implementação das regras de negócio para a normalização e padronização dos dados coletados.
1ª Quinzena de Setembro	<ul style="list-style-type: none"> Desenvolvimento da interface de busca principal do sistema. Implementação dos filtros avançados (marca, modelo, ano, preço, etc.). 	<ul style="list-style-type: none"> Desenvolvimento da visualização unificada dos anúncios. Criação da página de detalhes dos veículos agregados.
1ª Quinzena de Outubro	<ul style="list-style-type: none"> Implementação do enriquecimento de dados (exibição de Tabela FIPE). Desenvolvimento do algoritmo para identificação de anúncios duplicados. 	<ul style="list-style-type: none"> Implementação da funcionalidade de análise de preços de mercado para o Vendedor. Configuração da autenticação de usuários e desenvolvimento das interfaces de login e registro (para permitir futuras funcionalidades).
2ª Quinzena de Outubro	<ul style="list-style-type: none"> Implementação do painel de administração para monitorar a coleta de dados e a saúde do sistema. Desenvolvimento da interface para validação e moderação de conteúdo (remoção de duplicatas). 	<ul style="list-style-type: none"> Implementação de ferramentas de monitoramento (Grafana, Elastic Stack) para o backend. Implementação de testes unitários para as funcionalidades principais.
1ª Quinzena de Novembro	<ul style="list-style-type: none"> Implementação de melhorias de desempenho (caching, otimizações de banco de dados). Configuração de dashboards e alertas para o monitoramento contínuo. 	<ul style="list-style-type: none"> Realização de testes de aceitação para garantir que o MVP atende aos requisitos do projeto. Correções de bugs identificados nos testes e na etapa de desenvolvimento.
2ª Quinzena de Novembro	<ul style="list-style-type: none"> Desenvolvimento da estrutura inicial para a futura funcionalidade de anúncios próprios (modelos de dados e endpoints). Criação de dashboards e relatórios para a visualização das métricas do sistema. 	<ul style="list-style-type: none"> Implementação de feedback e avaliações de serviços (para futuras interações entre usuários). Realização de testes de carga e desempenho.
1ª Quinzena	<ul style="list-style-type: none"> Deploy da aplicação em 	<ul style="list-style-type: none"> Revisão e refinamento final do

de Dezembro	<p>ambiente de produção.</p> <ul style="list-style-type: none">• Containerização da aplicação (Docker) e preparação para o deploy final.	<p>código.</p> <ul style="list-style-type: none">• Verificação final do sistema em produção.• Criação e atualização da documentação técnica e de usuário (Swagger, Markdown).
--------------------	--	--

Tabela 4. Cronograma de atividades do projeto