

Desenvolvimento de um Sistema Web de Gerenciamento de Locações de Carros Utilizando o Framework CodeIgniter com PHP

Diego Rafael Muller
Guilherme Massinhani de Souza
Luis Ricardo Holscher

Resumo

O presente artigo descreve o desenvolvimento de um sistema web para o gerenciamento de locações de carros, utilizando o framework CodeIgniter com a linguagem de programação PHP. A aplicação foi concebida para otimizar e automatizar os processos de registro de veículos, clientes e locações, bem como o controle de disponibilidade da frota e a geração de relatórios gerenciais. Adotando o padrão arquitetural MVC (Model-View-Controller), o projeto visa garantir organização, modularidade e reutilização de código, facilitando a manutenção e a escalabilidade. Para o armazenamento dos dados, utilizou-se o banco de dados MySQL, integrado de forma eficiente ao framework. Os resultados obtidos demonstram a viabilidade da solução proposta, evidenciando o CodeIgniter como uma ferramenta robusta e ágil para o desenvolvimento de aplicações web no contexto de gerenciamento de locações de veículos.

Palavras-chave: CODEIGNITER. PHP. LOCAÇÃO. CARROS. SISTEMA. WEB.

Introdução

O setor de locação de veículos tem experimentado um crescimento contínuo, impulsionado pela flexibilidade e conveniência que oferece a indivíduos e empresas. Com essa expansão, a complexidade na gestão de frotas, clientes, contratos e manutenções aumenta exponencialmente. O gerenciamento manual desses processos torna-se propenso a erros, ineficiente e inviável em larga escala, resultando em perdas financeiras e insatisfação do cliente. Diante desse cenário, a necessidade de sistemas informatizados robustos e eficientes para gerenciar locadoras de carros é premente.

Este artigo propõe o desenvolvimento de um sistema web abrangente para o gerenciamento de locações de carros, utilizando a linguagem de programação PHP e o framework CodeIgniter. A escolha dessas tecnologias visa proporcionar uma solução ágil, segura e de fácil manutenção, capaz

de automatizar as operações diárias de uma locadora, desde o cadastro de veículos e clientes até o controle de locações e a geração de relatórios. O sistema busca otimizar a administração da frota, melhorar a experiência do cliente e reduzir a carga de trabalho operacional.

O objetivo principal deste trabalho é apresentar o processo de concepção, desenvolvimento e implementação de um sistema web de gerenciamento de locações de carros, detalhando as tecnologias empregadas, a arquitetura adotada e as funcionalidades implementadas. Serão abordados os desafios encontrados e as soluções aplicadas, bem como os resultados obtidos com a aplicação. A estrutura deste artigo está organizada da seguinte forma: a seção 2, "Fundamentação Teórica", discute os conceitos e tecnologias que servem de base para o projeto; a seção 3, "Desenvolvimento", detalha o processo de implementação do sistema, incluindo a modelagem do banco de dados e as funcionalidades; a seção 4, "Resultados", analisa o desempenho e a eficácia da solução; e, por fim, a seção 5, "Conclusão", sumariza as contribuições do trabalho e sugere futuras melhorias.

2 Fundamentação Teórica

2.1. Desenvolvimento Web

O desenvolvimento web refere-se à criação de aplicações que podem ser acessadas por meio de navegadores de internet. Essas aplicações são compostas por duas partes principais: o frontend e o backend. O frontend é a parte da aplicação com a qual o usuário interage diretamente, sendo construído com tecnologias como HTML (HyperText Markup Language) para estruturação do conteúdo, CSS (Cascading Style Sheets) para estilização e JavaScript para interatividade. O backend, por sua vez, é responsável pela lógica de negócios, processamento de dados, comunicação com o banco de dados e autenticação de usuários, utilizando linguagens de programação como PHP, Python ou Java.

A arquitetura de aplicações web modernas frequentemente adota padrões de design para organizar o código e facilitar o desenvolvimento. Entre esses padrões, o MVC (Model-View-Controller) destaca-se por sua capacidade de separar as responsabilidades da aplicação em camadas distintas, promovendo a modularidade, a reutilização de código e a escalabilidade. Essa separação é crucial para projetos complexos, pois permite que diferentes equipes trabalhem em partes distintas do sistema sem interferir umas nas outras, além de simplificar a manutenção e a evolução da aplicação.

2.2. Linguagem PHP

PHP (Hypertext Preprocessor) é uma linguagem de script de código aberto, amplamente utilizada para o desenvolvimento web dinâmico. Criada em 1994 por Rasmus Lerdorf, o PHP é executado no lado

do servidor, o que significa que o código é processado no servidor antes de ser enviado ao navegador do cliente. Sua popularidade deriva de sua facilidade de aprendizado, vasta documentação, grande comunidade de desenvolvedores e compatibilidade com diversos sistemas operacionais e servidores web, como Apache e Nginx.

Entre as principais características do PHP, destacam-se a integração facilitada com uma variedade de bancos de dados relacionais, a capacidade de gerar conteúdo HTML dinamicamente e o suporte a múltiplos frameworks e bibliotecas que aceleram o processo de desenvolvimento. A linguagem evoluiu significativamente ao longo dos anos, incorporando recursos de programação orientada a objetos e melhorias de desempenho, tornando-a uma escolha robusta para a construção de aplicações web de todos os portes.

2.3. Framework CodeIgniter e sua Escolha

CodeIgniter é um framework de desenvolvimento web para PHP conhecido por sua leveza, velocidade e simplicidade. Ele segue o padrão arquitetural MVC, proporcionando uma estrutura organizada para o desenvolvimento de aplicações web. Diferente de outros frameworks mais robustos, o CodeIgniter possui uma pequena pegada, o que contribui para um desempenho superior e tempos de carregamento mais rápidos. Sua curva de aprendizado é considerada suave, tornando-o acessível para desenvolvedores com diferentes níveis de experiência.

As principais características do CodeIgniter incluem um conjunto completo de bibliotecas para tarefas comuns de desenvolvimento web (como acesso a banco de dados, envio de e-mails, validação de formulários), suporte a Query Builder para interação com o banco de dados de forma mais segura e intuitiva, e recursos de segurança integrados, como filtragem XSS e gerenciamento de sessões. A comunidade ativa e a documentação clara e abrangente são outros pontos fortes que facilitam o desenvolvimento e a resolução de problemas.

Para o desenvolvimento do sistema de gerenciamento de locações de carros, o CodeIgniter foi escolhido por diversas razões. Sua leveza e desempenho são cruciais para uma aplicação que precisa ser responsiva e eficiente. A adesão ao padrão MVC promove uma estrutura de código limpa e organizada, facilitando a manutenção e a colaboração. Além disso, a simplicidade e a rapidez no desenvolvimento que o framework oferece são ideais para a construção de uma solução robusta em um tempo otimizado, alinhando-se perfeitamente aos objetivos do projeto.

2.4. Banco de Dados MySQL

O MySQL é um sistema de gerenciamento de banco de dados relacional (SGBDR) de código aberto, amplamente utilizado em aplicações web devido à sua robustez, escalabilidade e alta performance.

Ele organiza os dados em tabelas, que são interligadas por meio de relacionamentos, permitindo a recuperação e manipulação eficiente das informações. O MySQL suporta a linguagem SQL (Structured Query Language), que é o padrão para interação com bancos de dados relacionais, possibilitando operações como inserção, seleção, atualização e exclusão de dados.

No contexto de um sistema de gerenciamento de locações de carros, o MySQL é fundamental para armazenar uma vasta gama de informações, incluindo dados de clientes, detalhes de veículos, registros de locações, informações de usuários do sistema e categorias de veículos. Sua capacidade de garantir a integridade referencial entre as tabelas, por meio de chaves primárias e estrangeiras, assegura a consistência e a confiabilidade dos dados. A compatibilidade nativa com PHP e a integração facilitada com frameworks como o CodeIgniter tornam o MySQL uma escolha natural e eficiente para este tipo de aplicação.

2.5. Padrão Arquitetural MVC

O padrão Model-View-Controller (MVC) é uma abordagem arquitetural que visa separar as responsabilidades de uma aplicação em três componentes principais, a fim de melhorar a organização do código, a manutenibilidade e a escalabilidade. Essa separação permite que diferentes partes da aplicação sejam desenvolvidas e modificadas de forma independente, minimizando o impacto de alterações em outras áreas do sistema.

- **Model (Modelo):** Representa os dados e a lógica de negócios da aplicação. Ele interage diretamente com o banco de dados, sendo responsável por armazenar, recuperar e manipular os dados. O Modelo também contém as regras de negócio e validações que garantem a integridade dos dados. Em um sistema de locação de carros, o Modelo incluiria classes para Clientes, Veículos, Locações, etc., com métodos para gerenciar suas respectivas informações.
- **View (Visão):** É a camada de apresentação da aplicação, responsável por exibir os dados ao usuário. A Visão não contém lógica de negócios; seu principal objetivo é renderizar a interface do usuário (UI) com base nos dados fornecidos pelo Controlador. No CodeIgniter, as Views são arquivos HTML com trechos de código PHP que exibem as informações de forma formatada, como formulários de cadastro, listas de veículos ou detalhes de locações.
- **Controller (Controlador):** Atua como um intermediário entre o Modelo e a Visão. Ele recebe as requisições do usuário, processa-as, interage com o Modelo para obter ou manipular dados e, em seguida, seleciona a Visão apropriada para exibir a resposta ao usuário. O Controlador gerencia o fluxo da aplicação, coordenando as ações e garantindo que as informações corretas sejam apresentadas. No CodeIgniter, cada funcionalidade (ex: gerenciar clientes, registrar locação) geralmente possui um Controlador dedicado.

O CodeIgniter adota o padrão MVC de forma nativa e o incentiva fortemente, o que contribui para a criação de aplicações bem estruturadas e de fácil compreensão. Essa abordagem facilita o desenvolvimento colaborativo e a manutenção do código a longo prazo.

2.6. Mapeamento Objeto-Relacional (ORM)

O Mapeamento Objeto-Relacional (ORM) é uma técnica de programação que permite aos desenvolvedores interagir com bancos de dados relacionais utilizando objetos da linguagem de programação, em vez de escrever consultas SQL diretamente. O ORM atua como uma ponte entre o paradigma de programação orientada a objetos e o paradigma relacional dos bancos de dados, traduzindo as operações de objeto em comandos SQL e vice-versa.

Embora o CodeIgniter não possua um ORM completo como o ActiveRecord do Yii ou Eloquent do Laravel, ele oferece um poderoso Query Builder. O Query Builder do CodeIgniter fornece uma camada de abstração para o banco de dados, permitindo que as operações de CRUD (Create, Read, Update, Delete) sejam realizadas de forma mais orientada a objetos e segura, sem a necessidade de escrever SQL puro. Ele permite construir consultas complexas de maneira programática, o que é altamente benéfico para a produtividade e a segurança do desenvolvimento de aplicações web.

3. Desenvolvimento

3.1. Justificativa Tecnológica

A escolha das tecnologias para o desenvolvimento do sistema foi pautada na busca por uma solução que aliasse eficiência, segurança, facilidade de manutenção e agilidade no desenvolvimento. A linguagem PHP, com sua vasta comunidade e maturidade, oferece um ambiente robusto para aplicações web. O framework CodeIgniter foi selecionado por sua leveza e desempenho, características essenciais para um sistema que gerenciará um volume considerável de dados e transações. Sua aderência ao padrão MVC facilita a organização do código e a colaboração entre desenvolvedores. Por fim, o MySQL foi escolhido como sistema de gerenciamento de banco de dados devido à sua confiabilidade, escalabilidade e compatibilidade nativa com PHP e CodeIgniter, garantindo a persistência e integridade dos dados do sistema.

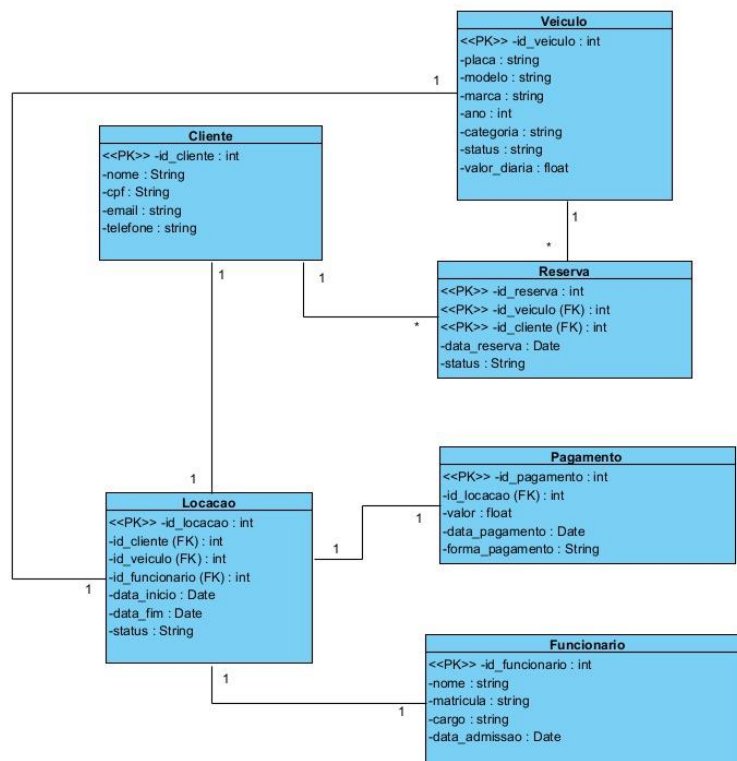
3.2. Modelagem do Banco de Dados

A modelagem do banco de dados foi projetada para representar as entidades essenciais do sistema de locação de carros e seus relacionamentos, garantindo a integridade e a consistência dos dados. As tabelas principais e seus atributos foram definidos para suportar as funcionalidades de

cadastro e gerenciamento. A seguir, são apresentadas as tabelas e seus respectivos campos:

- **cliente:** Armazena os dados dos clientes que utilizam o serviço de locação (id_cliente, nome, cpf, email, telefone)
- **veiculo:** Contém as informações dos veículos disponíveis para locação (id_veiculo, placa, modelo, marca, ano, categoria, status, valor_diaria)
- **reserva:** Registra as reservas feitas pelos clientes (id_reserva, id_veiculo, id_cliente, data_reserva, status)
- **locacao:** Representa as locações efetivamente realizadas (id_locacao, id_cliente, id_veiculo, id_funcionario, data_inicio, data_fim, status)
- **pagamento:** Registra os pagamentos realizados pelas locações (id_pagamento, id_locacao, valor, data_pagamento, forma_pagamento)
- **funcionário:** Contém os dados dos funcionários da empresa (id_funcionario, nome, matricula, cargo, data_admissao)

Um Diagrama Entidade-Relacionamento (ER) detalhado seria apresentado para ilustrar visualmente as entidades, seus atributos e os relacionamentos entre as tabelas, incluindo chaves primárias e estrangeiras.



3.3. Integração com o Banco via ORM

No CodeIgniter, a interação com o banco de dados é facilitada pelo seu poderoso Query Builder, que atua como uma camada de abstração, permitindo a manipulação de dados de forma mais segura e orientada a objetos, similar a um ORM. Isso elimina a necessidade de escrever SQL puro na maioria das operações, reduzindo a chance de erros e vulnerabilidades como SQL Injection. O Query Builder permite construir consultas complexas de forma programática, como inserções, atualizações, exclusões e seleções de dados.

Por exemplo, para adicionar um novo cliente, o processo envolve a criação de um array com os dados e a utilização do método `insert()` do Query Builder. Da mesma forma, para atualizar o status de um veículo ou buscar locações ativas, métodos como `update()` e `get()` são empregados, respectivamente. Essa abordagem não apenas acelera o desenvolvimento, mas também torna o código mais legível e fácil de manter, pois as operações de banco de dados são representadas por métodos intuitivos do framework.

3.4. Funcionalidades Implementadas

O sistema de gerenciamento de locações de carros foi desenvolvido com um conjunto de funcionalidades essenciais para otimizar as operações de uma locadora. As principais funcionalidades implementadas são:

3.4.1. Cadastro e Gerenciamento

- **Clientes:** Módulo completo para adicionar, atualizar, listar e desativar clientes. Cada cliente possui informações como nome, CPF, telefone e endereço. A desativação permite manter o histórico sem remover permanentemente os dados.
- **Veículos:** Permite o cadastro detalhado de veículos, incluindo marca, modelo, ano, placa, cor, e a associação a uma categoria específica. O status do veículo (disponível, locado, manutenção) é controlado para refletir sua condição atual na frota.
- **Categorias de Veículos:** Funcionalidade para definir e gerenciar diferentes categorias de veículos (ex: compacto, SUV, luxo), cada uma com sua diária padrão e descrição. Isso permite uma precificação flexível e organizada.
- **Usuários do Sistema:** Controle de acesso para usuários internos
- (administradores e operadores) com sistema de login, senhas criptografadas e a diferenciação de níveis de acesso para garantir a segurança e a segregação de responsabilidades.

3.4.2. Locação de Veículos

- **Registro de Locação:** Permite registrar novas locações, incluindo a data de retirada, a data de devolução prevista e, posteriormente, a data de devolução efetiva. O sistema associa a locação a um cliente e a um veículo específico.
- **Cálculo Automático de Valor:** O sistema calcula automaticamente o valor total da locação com base na diária da categoria do veículo e no período de locação. Isso minimiza erros manuais e agiliza o processo.
- **Aplicação de Multa por Atraso:** Caso a devolução efetiva ocorra após a data prevista, o sistema é capaz de calcular e aplicar multas por atraso, conforme regras predefinidas.
- **Controle de Status da Locação:** O status de cada locação é atualizado automaticamente (ativa, finalizada, cancelada), fornecendo uma visão clara do andamento das operações.
- **Atualização de Status do Veículo:** De forma integrada, o status do veículo é automaticamente atualizado (ex: de “disponível” para “locado” no início da locação, e de “locado” para “disponível” ou “manutenção” na devolução).

3.4.3. Acesso e Segurança

- **Sistema de Login:** Implementação de um sistema de login seguro com autenticação de usuários por meio de senhas criptografadas (utilizando algoritmos de hash seguros) para proteger as credenciais de acesso.
- **Diferenciação de Permissões:** O sistema distingue entre diferentes níveis de acesso (administrador e operador), garantindo que cada usuário tenha acesso apenas às funcionalidades e dados pertinentes ao seu perfil, reforçando a segurança e a integridade do sistema.

3.4.4. Organização e Desempenho

- **Otimização de Consultas:** Foram criados índices em campos frequentemente utilizados em consultas (como placa de veículo, CPF de cliente e status de veículo/locação) para melhorar significativamente a performance das operações de busca e filtragem de dados.
- **Controle de Criação e Atualização:** A inclusão dos campos `created_at` e `updated_at` em todas as tabelas principais permite o rastreamento automático da data e hora de criação e da última modificação de cada registro, facilitando a auditoria e o gerenciamento de dados.

3.5. Etapas do Desenvolvimento

O desenvolvimento do sistema seguiu uma abordagem iterativa e incremental, dividida nas seguintes fases:

- **Levantamento de Requisitos:** Nesta fase inicial, foram identificadas e documentadas as necessidades e expectativas dos usuários, resultando na definição das funcionalidades essenciais e não essenciais do sistema. Isso incluiu a coleta de informações sobre os processos de locação, cadastro de clientes e veículos, e requisitos de segurança.
- **Modelagem:** Com base nos requisitos levantados, foi realizada a modelagem do banco de dados, com a criação do Diagrama Entidade-Relacionamento (ER) e a definição das tabelas, atributos e relacionamentos. Paralelamente, a arquitetura do sistema foi desenhada, seguindo os princípios do padrão MVC.
- **Configuração do Ambiente:** Esta etapa envolveu a instalação e configuração do framework CodeIgniter, a configuração da conexão com o banco de dados MySQL e a instalação de quaisquer bibliotecas ou dependências adicionais necessárias para o projeto.
- **Implementação:** A fase de implementação consistiu no desenvolvimento dos módulos do sistema (autenticação, gerenciamento de veículos, clientes, categorias e locações), seguindo o padrão MVC. As Views foram construídas utilizando HTML, CSS e JavaScript, enquanto os Controllers e Models foram desenvolvidos em PHP, utilizando o Query Builder do CodeIgniter para a interação com o banco de dados.
- **Testes e Ajustes:** Após a implementação de cada módulo, foram realizados testes unitários e de integração para verificar a correção das funcionalidades e a interação entre os componentes. Quaisquer bugs ou inconsistências foram identificados e corrigidos, e otimizações de desempenho foram aplicadas para garantir a fluidez do sistema.

4. Resultados

Após a conclusão do processo de desenvolvimento, o sistema web de gerenciamento de locações de carros demonstrou ser uma solução funcional e eficiente, atendendo aos requisitos levantados e proporcionando uma melhoria significativa nos processos de gestão de uma locadora. A aplicação foi testada em um ambiente simulado, e os resultados obtidos confirmam a viabilidade e a robustez da solução.

O sistema oferece uma interface intuitiva e de fácil navegação, permitindo que os usuários (administradores e operadores) realizem as operações de cadastro, gerenciamento e locação de forma ágil e sem a necessidade de conhecimento técnico aprofundado. As funcionalidades de cadastro de clientes, veículos e categorias, bem como o controle de locações, foram implementadas com sucesso, garantindo a integridade e a consistência dos dados.

Um dos principais benefícios alcançados foi a automação do processo de locação, desde o cálculo automático do valor total até a atualização do status do veículo. Isso resultou em uma redução considerável de erros manuais e no aumento da eficiência operacional. A implementação de índices nas tabelas do banco de dados, como `plate` em `vehicles` e `cpf` em `clients`, contribuiu para uma performance otimizada nas consultas, mesmo com um volume crescente de dados.

O uso do framework CodeIgniter e a adesão ao padrão MVC resultaram em um código bem organizado, modular e de fácil manutenção. A separação clara entre Model, View e Controller facilitou o desenvolvimento colaborativo e a identificação e correção de eventuais problemas. A segurança foi reforçada com a implementação de um sistema de login com senhas criptografadas e a diferenciação de permissões por nível de acesso, protegendo as informações sensíveis do sistema.

Embora o projeto tenha enfrentado desafios comuns ao desenvolvimento de software, como a integração de diferentes módulos e a otimização de consultas, a abordagem iterativa e a escolha de tecnologias adequadas permitiram superar esses obstáculos de forma eficaz. A experiência adquirida durante o desenvolvimento do sistema proporcionou um aprendizado prático significativo sobre engenharia de software, desenvolvimento web com frameworks modernos e boas práticas de segurança.

5. Conclusão

Este artigo apresentou o desenvolvimento de um sistema web para o gerenciamento de locações de carros, utilizando o framework CodeIgniter com PHP e o banco de dados MySQL. O projeto demonstrou a viabilidade de construir uma aplicação robusta e eficiente, capaz de automatizar e otimizar os processos de uma locadora de veículos, desde o cadastro e gerenciamento de clientes e veículos até o controle de locações e a geração de relatórios.

A escolha do CodeIgniter mostrou-se acertada, dada a sua leveza, desempenho e a adesão ao padrão MVC, que proporcionou uma estrutura de código organizada e de fácil manutenção. A integração com o MySQL, utilizando o Query Builder do CodeIgniter, garantiu a segurança e a eficiência na manipulação dos dados. As funcionalidades implementadas, como o cálculo automático de valores e a atualização de status, contribuem significativamente para a redução de erros e o aumento da produtividade.

Como trabalhos futuros, sugere-se a expansão do sistema com a integração de módulos de pagamento online, a implementação de um módulo de manutenção de veículos mais detalhado, a criação de um aplicativo móvel para clientes e a incorporação de inteligência artificial para análise preditiva de demanda e otimização de preços. Acreditamos que este trabalho serve como um guia prático para o desenvolvimento de sistemas de gerenciamento similares, evidenciando o potencial do CodeIgniter para a criação de soluções web eficazes e escaláveis.

6. Referências

CODEIGNITER. *Welcome to CodeIgniter*. Disponível em: <https://www.codeigniter.com/>. Acesso em: 24 jun. 2025.

GEEKHUNTER. *Boas práticas de programação em PHP*. Disponível em: <https://blog.geekhunter.com.br/boas-praticas-de-programacao-em-php/>. Acesso em: 29 jun. 2025.

IONOS. *PHP Framework CodeIgniter*. Disponível em: <https://www.ionos.com/digitalguide/websites/web-development/codeigniter-the-lean-php-framework/>. Acesso em: 25 jun. 2025.

IUVEN ERP. *Sistema para locadora de veículos: veja funcionalidades*. Disponível em: <https://iuven.com.br/sistema-para-locadora-de-veiculos-veja-funcionalidades/>. Acesso em: 28 jun. 2025.

KODEZI. *What Are MVC Frameworks for PHP? Understanding the Basics and Benefits*. Disponível em: <https://blog.kodezi.com/what-are-mvc-frameworks-for-php-understanding-the-basics-and-benefits/>. Acesso em: 29 jun. 2025.

LOC1. *Software de Gestão para locadoras de Veículos e Mobilidade*. Disponível em: <https://loc1.com.br/>. Acesso em: 27 jun. 2025.

RENTSYST. *Sobre RentSyst - Sistema de aluguel de automóveis*. Disponível em: <https://www.rentsyst.com/pt/about>. Acesso em: 29 jun. 2025.

STUDOCU. *Sistema de locação de carros (Diagramas UML e Requisitos)*. Disponível em: <https://www.studocu.com/pt-br/document/universidade-paulista/tec-em-analise-e-desenvolvimento-de-sistemas/sistema-de-locacao-de-carros-diagramas-uml-e-requisitos/58699593>. Acesso em: 29 jun. 2025.