**Protocolos e Segurança em Redes WiFi**: Segurança aplicada a redes WiFi

Gabriela Mendes Demossi 1

## Resumo

Este artigo apresenta o desenvolvimento de um sistema web para gerenciamento de caminhões e controle de usuários em empresas, com foco na organização e segurança dos dados relacionados à frota. O sistema foi construído utilizando o framework CodeIgniter em PHP, com banco de dados MySQL e aplicação dos conceitos de Programação Orientada a Objetos (OO) e arquitetura MVC. A solução permite cadastrar, visualizar, editar e excluir caminhões, além de gerenciar usuários com autenticação segura. Foram elaborados diagramas de casos de uso, sequência, classes e modelo entidade-relacionamento para fundamentar a modelagem do sistema. O projeto buscou atender à necessidade de centralização e controle eficiente de informações, reduzindo erros e otimizando a gestão da frota. Ao final, destaca-se o aprendizado obtido sobre modelagem, implementação de sistemas seguros e práticas de versionamento com Git, evidenciando a importância da integração entre teoria e prática no desenvolvimento de software.

**Palavras-chave:** Gestão de Frotas; Controle de Caminhões; CodeIgniter; Modelagem de Software; Banco de Dados.

# 1 INTRODUÇÃO

A gestão de frotas de caminhões é um desafio constante para empresas com médio e grande volume de veículos, exigindo controle rigoroso sobre informações como documentos, informações de seguro e dados dos veículos. A falta de um sistema informatizado pode acarretar erros manuais, perda de prazos e dificuldades na organização, prejudicando a eficiência operacional.

Este trabalho apresenta o desenvolvimento de um sistema web destinado ao **controle de caminhões e gerenciamento de usuários**, visando centralizar e facilitar o registro de dados da frota, além de implementar um mecanismo seguro de autenticação de usuários. O sistema foi construído utilizando o framework CodeIgniter em PHP, explorando os conceitos de programação orientada a objetos (OO) e arquitetura MVC para estruturar a aplicação de forma modular e de fácil manutenção.

Durante o processo de desenvolvimento, foram elaborados artefatos de modelagem como diagramas de casos de uso, sequência, classes e modelo entidade-relacionamento (ER), garantindo a coerência entre a modelagem e a implementação prática do sistema. Este artigo detalha as etapas do projeto, as tecnologias utilizadas, os fundamentos teóricos aplicados e as decisões tomadas, buscando demonstrar a importância da integração entre planejamento e execução para a construção de sistemas robustos e funcionais.

# 2 DESENVOLVIMENTO

## 2.1 Fundamentação Técnica

O sistema foi construído com base em princípios de Programação Orientada a Objetos (OO), permitindo modularidade, reuso de código e fácil manutenção. A arquitetura MVC (Model-View-Controller) foi aplicada utilizando o framework CodeIgniter, separando as camadas de dados, lógica de negócios e apresentação para garantir organização e escalabilidade.

Para garantir segurança, o sistema implementou **autenticação de usuários** com criptografia de senhas utilizando password_hash() e password_verify(), evitando o armazenamento de senhas em texto plano. Além disso, o controle de sessões foi configurado para restringir o acesso às funcionalidades apenas a usuários autenticados.

No aspecto de **versionamento**, foi utilizado o Git em conjunto com a plataforma GitHub, permitindo o acompanhamento de alterações, colaboração e recuperação de versões anteriores do projeto, seguindo boas práticas de desenvolvimento profissional.

## 2.2 Metodologias de Desenvolvimento

A metodologia adotada foi incremental, com ciclos curtos de planejamento, implementação e testes para cada funcionalidade. Inicialmente, foram definidos os requisitos e elaborada a modelagem do sistema por meio de diagramas UML, garantindo que as funcionalidades estivessem claramente especificadas. Em seguida, as funcionalidades foram

desenvolvidas e testadas isoladamente, com validações intermediárias e ajustes constantes, até a construção do sistema completo.

Essa abordagem permitiu a identificação e correção de problemas rapidamente, além de possibilitar melhorias durante o processo de desenvolvimento.

## 2.3 Tecnologias e Frameworks Utilizados

- **PHP 7.4+**: linguagem principal para o desenvolvimento do back-end do sistema.
- **CodeIgniter 4.x**: Por ser uma ferramenta leve, de fácil configuração e altamente aderente ao padrão arquitetural **MVC (Model-View-Controller)**, o que facilita a separação de responsabilidades no código e melhora a manutenção do projeto.
- **MySQL**: sistema gerenciador de banco de dados relacional, utilizado para armazenar as informações dos usuários e caminhões.
- **HTML, CSS e JavaScript**: tecnologias utilizadas no front-end para a criação das interfaces do sistema.
- **Git e GitHub**: ferramentas para versionamento de código, controle de alterações e colaboração.
- **Draw.io**: ferramenta para criação dos diagramas de modelagem (casos de uso, sequência, classes e ER).

## 2.4 Etapas do Processo Construtivo

- Modelagem

Foram elaborados diagramas de Casos de Uso, Sequência, Classes e Entidade-Relacionamento (ER), que serviram como base para organizar as funcionalidades do sistema e estruturar o banco de dados.

- **Diagrama de Casos de Uso**

O diagrama ilustra claramente as ações disponíveis para os dois atores: Admin (com acesso completo, incluindo cadastro de usuários) e Usuário (restrito a ações sobre caminhões). As funcionalidades incluem login, visualização de caminhões, cadastro, edição, exclusão de caminhões e logout. Para o Admin, inclui também o cadastro de novos usuários.

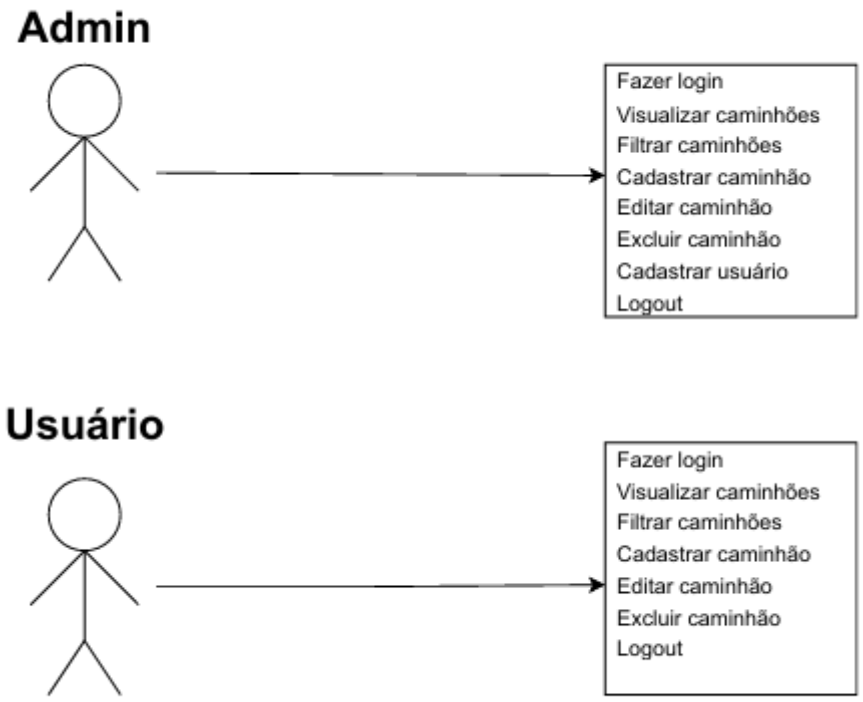



*Diagrama de Casos de Uso*

- **Diagrama de Sequência**

O diagrama de sequência representa fluxos críticos do sistema: login, visualização de caminhões e cadastro de caminhões. Ele demonstra a interação entre os objetos Usuário/Admin, Sistema Web e Banco de Dados, evidenciando a ordem das chamadas de métodos e retornos.

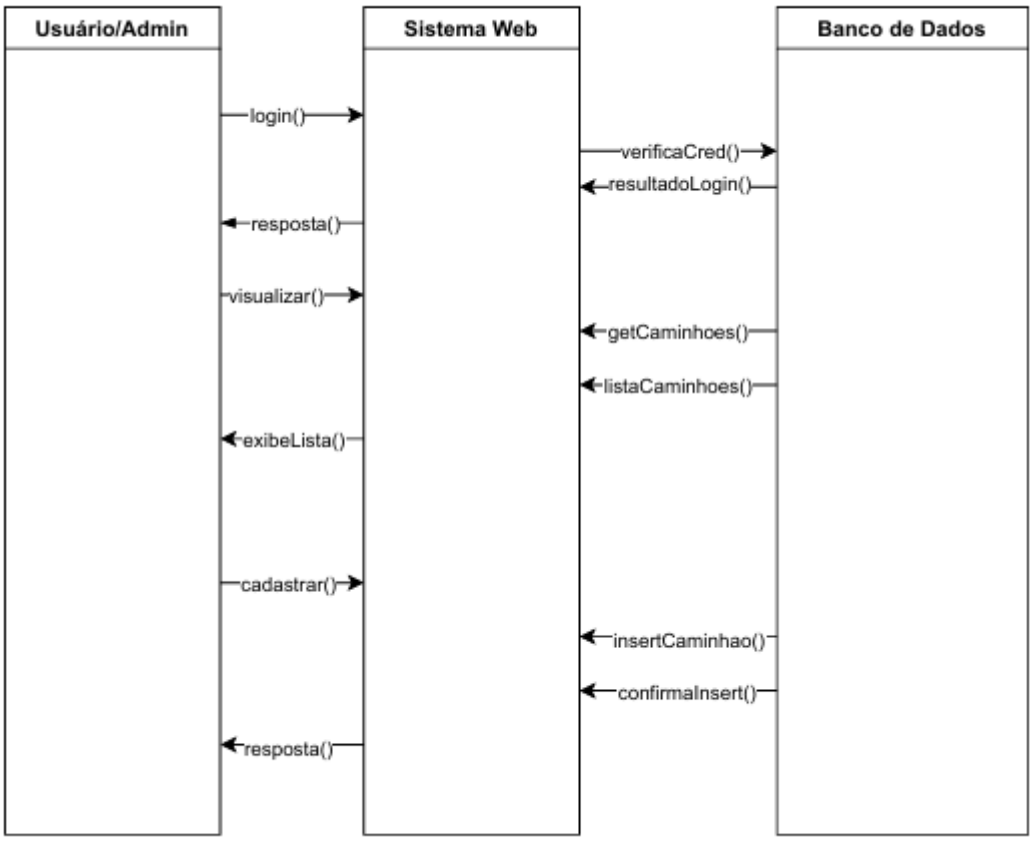



*Diagrama de Sequência*

- **Diagrama de Classes**

O diagrama define as entidades Usuario, Caminhao, Sistema e Banco de Dados. Cada classe apresenta seus atributos e métodos principais, mostrando os relacionamentos e responsabilidades. A modelagem reflete a estrutura do código implementado, que segue o padrão MVC do CodeIgniter.

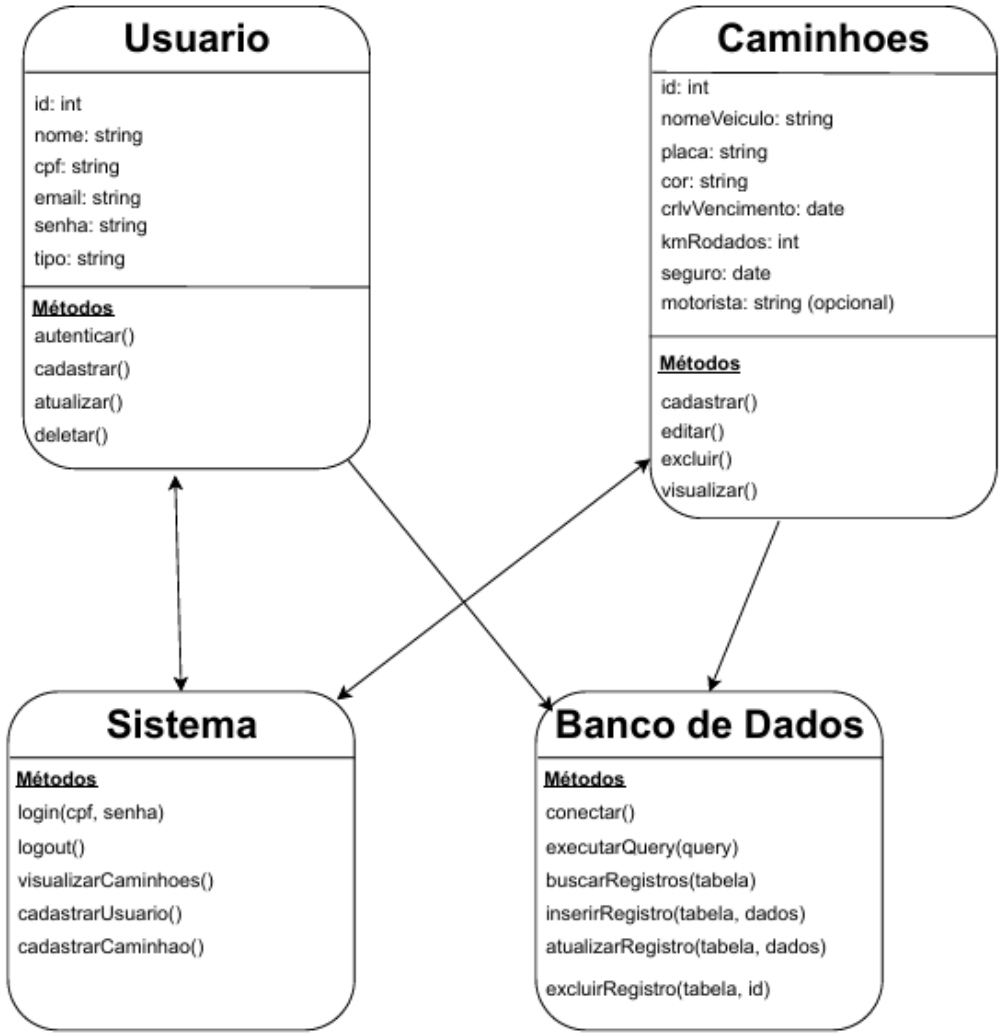
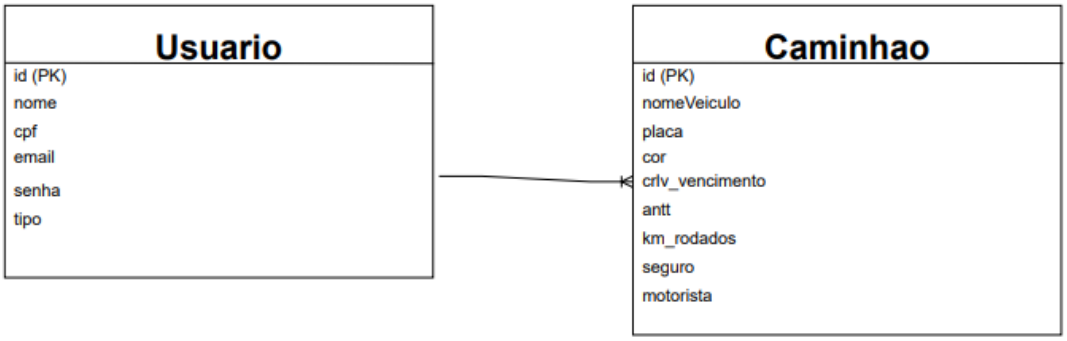



*Diagrama de Classes*

- **Modelo Entidade-Relacionamento (ER)**

O modelo ER apresenta as tabelas principais do banco: usuarios e caminhoes, com seus campos e relacionamentos. Ele demonstra como as entidades do sistema estão persistidas no banco de dados MySQL.




*Modelo Entidade-Relacionamento (ER)*

- Implementação

A construção do sistema iniciou pelas rotas de autenticação de usuários e pelo CRUD de caminhões, com foco em permitir que usuários cadastrassem, visualizassem, editassem e excluíssem caminhões. Posteriormente, foi desenvolvido o módulo de gerenciamento de usuários, disponível apenas para administradores.

- Testes

Foram realizados testes manuais de cada funcionalidade, validando o comportamento esperado, verificando fluxos alternativos (ex.: login com senha errada) e corrigindo eventuais falhas.

- Deploy

O sistema foi hospedado em ambiente local utilizando XAMPP, configurando o banco de dados MySQL e integrando o projeto com o servidor Apache. Além disso, o código-fonte foi disponibilizado em repositório GitHub, garantindo acesso ao histórico de desenvolvimento e facilitando futuras atualizações.

## 3 CONCLUSÃO

O desenvolvimento deste sistema web para controle de caminhões e gerenciamento de usuários possibilitou a construção de uma solução prática, funcional e segura para empresas que necessitam gerenciar informações de frota de forma centralizada. O uso de conceitos como **Programação Orientada a Objetos** e o padrão arquitetural **MVC** foram fundamentais para garantir a organização e a manutenibilidade do projeto, permitindo a separação clara entre dados, lógica de negócios e interface.

A elaboração prévia dos **diagramas de modelagem** (casos de uso, sequência, classes e entidade-relacionamento) demonstrou-se essencial para guiar a implementação e garantir coerência entre o planejado e o produto final, além de facilitar a identificação de melhorias durante o processo de desenvolvimento.

Durante as etapas do projeto, foi possível aplicar práticas importantes como **autenticação segura**, controle de acesso, uso de **versionamento com Git** e boas práticas de codificação, reforçando a importância da segurança e da organização no desenvolvimento de sistemas.

Entre os principais aprendizados, destacam-se a importância de alinhar a modelagem com a implementação, a aplicação de padrões de projeto para construir sistemas robustos e a experiência prática com ferramentas de versionamento e frameworks modernos, que são essenciais para o mercado de trabalho. O projeto reforçou ainda mais a relevância da integração entre conceitos teóricos e prática no processo de construção de software.

## Abstract

This article presents the development of a web system for truck management and user control in companies, focusing on the organization and security of fleet-related data. The system was built using the CodeIgniter framework in PHP, with a MySQL database and application of Object-Oriented Programming (OO) and MVC architecture concepts. The solution allows registering, viewing, editing and deleting trucks, in addition to managing users with secure authentication. Use case, sequence, class and entity-relationship model diagrams were developed to support the system modeling. The project sought to meet the need for centralization and efficient control of information, reducing errors and optimizing fleet management. At the end, the learning obtained about modeling, implementation of secure systems and versioning practices with Git is highlighted, highlighting the importance of integrating theory and practice in software development.

**Keywords**: Fleet Management; Truck Control; CodeIgniter; Software Modeling; Database.