

**ETEC JD. ÂNGELA**

**CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

**ADRIANA COSTA DOS SANTOS**

**CARLOS EDUARDO CRISPIM SANTANA**

**LUCAS SILVA DOS SANTOS**

**PEDRO HENRIQUE SANTOS SILVA**

**RAV**

**REGISTRO DE ACESSO DE VEÍCULOS**

**SÃO PAULO**

**2026**

ADRIANA COSTA DOS SANTOS

CARLOS EDUARDO CRISPIM SANTANA

LUCAS SILVA DOS SANTOS

PEDRO HENRIQUE SANTOS SILVA

**RAV**

## **REGISTROS DE ACESSO DE VEÍCULOS**

Trabalho de Conclusão de Curso apresentado à Etec Jardim Ângela, do Centro Estadual de Educação Tecnológica Paula Souza, como requisito para a obtenção do diploma de Técnico de Nível Médio em Desenvolvimento de Sistemas sob a orientação do Professor Especialista Éricles Pinheiro Ferreira.

**SÃO PAULO**

**2026**

## SUMÁRIO

|                                               |    |
|-----------------------------------------------|----|
| 1. INTRODUÇÃO.....                            | 5  |
| 2. JUSTIFICATIVA.....                         | 5  |
| 3. PROBLEMA.....                              | 6  |
| 4. HIPÓTESE.....                              | 7  |
| 5. OBJETIVOS.....                             | 7  |
| 6. OBJETIVOS ESPECÍFICOS.....                 | 7  |
| 7. METODOLOGIA DA PESQUISA.....               | 8  |
| 8. FUNDAMENTAÇÃO TEÓRICA .....                | 8  |
| 9. METODOLOGIA DO TRABALHO .....              | 9  |
| 10. INSTRUMENTOS DE COLETA DE DADOS .....     | 9  |
| 11. ANÁLISE DOS DADOS .....                   | 10 |
| 12. FERRAMENTAS E LINGUAGENS UTILIZADAS ..... | 11 |
| 13. CRONOGRAMA DO TRABALHO .....              | 18 |
| 14. DOCUMENTAÇÃO DO SOFTWARE .....            | 18 |
| 15. FLUXOGRAMA .....                          | 25 |
| 16. CAPTURA DE TELAS .....                    | 26 |
| 17. DIAGRAMA DE CASO DE USO.....              | 39 |
| 18. MODELO CONCEITUAL DO BANCO DE DADOS ..... | 40 |
| 19. MODELO LÓGICO DO BANCO DE DADOS .....     | 41 |
| 20. MODELO FÍSICO DO BANCO DE DADOS .....     | 42 |
| 21. CICLO DE VIDA DO PROGRAMA .....           | 49 |
| 22. CONSIDERAÇÕES FINAIS.....                 | 51 |
| 23. REFERÊNCIAS .....                         | 52 |

## LISTA DE IMAGENS

|                                                               |    |
|---------------------------------------------------------------|----|
| Figura 1 - Fluxograma .....                                   | 25 |
| Figura 2 - Homepage .....                                     | 26 |
| Figura 3 - Código da homepage.....                            | 26 |
| Figura 4 - Cadastro .....                                     | 27 |
| Figura 5 - Código do cadastro .....                           | 27 |
| Figura 6 - Login .....                                        | 28 |
| Figura 7 - Código do login .....                              | 28 |
| Figura 8 - Painel Operacional.....                            | 29 |
| Figura 9 - Código do painel operacional.....                  | 29 |
| Figura 10 - Cadastro de usuários e automóveis.....            | 31 |
| Figura 11 - Código do cadastro de usuários e automóveis ..... | 31 |
| Figura 12 - Acessos .....                                     | 33 |
| Figura 13 - Código de acessos.....                            | 33 |
| Figura 14- Estacionamento .....                               | 34 |
| Figura 15 - Código do estacionamento .....                    | 34 |
| Figura 16 - Gestão de cadastro.....                           | 35 |
| Figura 17 - Código da gestão de cadastro .....                | 35 |
| Figura 18 - Relatório.....                                    | 36 |
| Figura 19 - Código do relatório.....                          | 37 |
| Figura 20 - Configurações .....                               | 37 |
| Figura 21 - Código das configurações.....                     | 38 |
| Figura 22 - Diagrama de caso de uso .....                     | 39 |
| Figura 23 - Diagrama de caso de uso .....                     | 40 |
| Figura 24 - Modelo lógico do banco de dados.....              | 41 |

## **1. INTRODUÇÃO**

O Registro de Acesso de Veículos (RAV) tem como objetivo principal modernizar e facilitar a entrada e saída de automóveis e motocicletas no perímetro escolar das unidades Etec. Através da substituição de métodos manuais por uma interface digitalizada, busca-se otimizar o tempo de atendimento nas portarias, manter a consistência dos dados, e consequentemente, elevar o nível de segurança da comunidade acadêmica.

## **2. JUSTIFICATIVA**

Atualmente, grande parte das unidades de ETECs e FATECs ainda gerencia o controle de acesso de seus perímetros de forma manual ou isolada, utilizando planilhas físicas locais. Esse modelo gera lentidão no fluxo diário de alunos, professores e funcionários, além de abrir margem para perda de dados e rasuras. O RAV surge para unificar e digitalizar essa operação, transformando a portaria escolar em um ambiente ágil, seguro e auditável.

O RAV justifica-se como uma resposta viável por não exigir infraestruturas de hardware caras ou embarcadas (como sensores RFID ou câmeras OCR), mantendo o foco em uma aplicação web fluida, econômica e de rápida adoção pela equipe operacional da guarita.

Justificar o trabalho é reconhecer que os métodos atuais são insuficientes, e que a digitalização melhora diretamente o desempenho operacional.

### 3. PROBLEMA

Desenvolver um sistema web para o controle de acesso de veículos, com o objetivo de substituir os registros manuais em papel ou planilhas. Buscar aumentar a eficiência do processo, reduzir erros humanos, diminuir o tempo de registro.

Implementar a solução como uma plataforma web utilizando tecnologias acessíveis e de desempenho adequado, tais como HTML5, CSS3, JavaScript, PHP e o sistema gerenciador de banco de dados MySQL. A arquitetura de autenticação e controle de privilégios do software é estruturada de forma simplificada e direta por unidade de ensino, eliminando o gerenciamento complexo de contas individuais. O sistema divide-se em duas chaves de acesso estratégicas: a **Chave Administrativa** (que concede privilégios elevados para auditoria e autenticação de configurações avançadas da ETEC/FATEC) e a **Chave de Portaria/Operacional** (destinada estritamente aos operadores da guarita para o registro ágil e controle diário do fluxo veicular).

Aplicar o sistema de forma customizada e exclusiva no ecossistema das unidades do Centro Paula Souza (ETECs e FATECs) que ainda dependem de triagens manuais ou analógicas para a contenção e registro perimetral.

Minimizar o impacto e o custo financeiro do projeto durante sua fase acadêmica e futura implantação institucional, projetando o sistema para rodar de forma autônoma em um servidor local gratuito (ambientes virtualizados como Laragon ou XAMPP). Essa abordagem elimina custos com hospedagem em nuvem (*cloud computing*), mantém os dados sensíveis da instituição protegidos localmente e anula os riscos de indisponibilidade de internet na guarita, limitando os custos do projeto apenas ao tempo investido em desenvolvimento técnico pela equipe.

## 4. HIPÓTESE

Parte-se da hipótese de que a substituição dos registros analógicos por uma plataforma web cliente-servidor, parametrizada para operação em servidor local e estruturada com chaves de acesso simplificadas (Administrativa e Portaria), otimizará entre 60% e 70% o tempo de triagem veicular nas guaritas das ETECs e FATECs.

## 5. OBJETIVOS

O objetivo geral deste trabalho é projetar e desenvolver o RAV, uma plataforma web em PHP e MySQL para o gerenciamento de acesso veicular em unidades do Centro Paula Souza. A solução foca na substituição do papel e de planilhas locais na guarita, garantindo maior organização e segurança através de uma arquitetura baseada em servidor local e chaves de acesso simplificadas, sem a necessidade de investimentos em hardwares caros ou infraestruturas complexas de rede.

## 6. OBJETIVOS ESPECÍFICOS

- **Facilitar** o registro de entradas e saídas de veículos de forma ágil e segura;
- **Implementar** o cadastro e gerenciamento de veículos, funcionários e acessos;
- **Permitir** a geração de relatórios detalhados, filtrados por período;
- **Reduzir** a dependência de registros manuais, minimizando erros;
- **Aumentar** a confiabilidade e a segurança das informações armazenadas.

## 7. METODOLOGIA DA PESQUISA

Combinar **pesquisa bibliográfica** e **pesquisa de campo** como metodologia para o desenvolvimento do projeto. Realizar a pesquisa bibliográfica por meio da análise de documentação direta e indireta, incluindo o estudo de softwares de mercado com funcionalidades similares, a fim de compreender as soluções existentes e definir o campo de aplicação do sistema.

Conduzir a pesquisa de campo por meio de **entrevistas semiestruturadas** com os potenciais usuários, buscando obter uma visão aprofundada de suas rotinas operacionais e das dificuldades enfrentadas. Documentar e utilizar os dados coletados em ambas as frentes como base para o levantamento de requisitos do sistema.

## 8. FUNDAMENTAÇÃO TEÓRICA

### Tecnologias e diferenciais competitivos

A escolha pelo desenvolvimento estruturado em plataforma web justifica-se por sua ampla acessibilidade e portabilidade, permitindo que o sistema seja operado a partir de qualquer dispositivo dotado de um navegador padrão, independentemente do sistema operacional ativo na guarita.

O projeto do RAV estabelece diferenciais estratégicos claros e intencionais quando comparado a soluções de mercado e trabalhos correlatos

- **Em Relação à Tecnologia de Identificação (Pimentel, 2023):** Diferente de propostas que exigem a complexidade de implantação de hardwares embarcados e protocolos físicos baseados em tecnologia RFID para o controle veicular, o RAV foca estritamente na camada de software e lógica web. Isso elimina custos de aquisição, infraestrutura e manutenção de componentes físicos e circuitos elétricos nas escolas.
- **Em Relação à Dependência Externa (Ferro e Bampa, 2021):** O projeto descarta qualquer dependência de APIs proprietárias ou aplicativos de

terceiros (como mensageiros instantâneos e integrações de delivery) para o envio de alertas ou relatórios. O foco reside na governança interna e autônoma dos dados dentro da própria instituição, blindando o fluxo contra instabilidades externas de rede.

- **Foco em Gestão Simplificada por Chaves:** O diferencial primário da solução atual é a substituição da burocracia de múltiplos perfis de usuários por um modelo de chaves institucionais unificadas por unidade. A segregação de privilégios opera através de uma **Chave de Portaria** (operacional) e uma **Chave Administrativa** (configurações elevadas e auditoria), projetada para reduzir em até 80% as falhas humanas de preenchimento e elevar em até 70% a eficiência na triagem veicular.

## 9. METODOLOGIA DO TRABALHO

Para o desenvolvimento do RAV, a abordagem metodológica adotada é a **qualitativa**: Concentra-se na interpretação dos fenômenos operacionais, rotinas diárias, percepções de segurança e significados atribuídos pelos operadores de guarita e funcionários administrativos em relação às vulnerabilidades dos métodos manuais tradicionais. Essa análise foi crucial para mapear a usabilidade da interface e a lógica de contingência do sistema.

## 10. INSTRUMENTOS DE COLETA DE DADOS

O principal instrumento de pesquisa de campo utilizado pelo grupo foi a **entrevista semiestruturada**, aplicada diretamente com os profissionais encarregados pela segurança patrimonial e controle de fluxo veicular nas portarias das unidades de ensino.

A escolha por este instrumento justifica-se por sua flexibilidade, permitindo que os entrevistados respondam às perguntas com base em sua experiência prática na

função e em suas opiniões reais sobre o processo analógico atual. O roteiro de perguntas foi elaborado para levantar os seguintes requisitos de software:

- O tempo médio gasto para coletar manualmente dados de placas e condutores em papel.
- Os principais motivos de lentidão e formação de filas nos horários de pico acadêmico.
- O comportamento e o histórico de incidentes operacionais decorrentes de quedas de energia ou ausência de conectividade de rede na guarita.

## 11. ANÁLISE DOS DADOS

Depois de realizarmos as entrevistas com as equipes que trabalham nas guaritas, organizamos as principais reclamações e dificuldades relatadas para definir como o RAV resolveria esses problemas no dia a dia da Etec/Fatec:

- **Identificação de Alunos e Agilidade:** A maior reclamação foi a lentidão e a complicação para identificar os alunos e registrar as placas dos veículos nos horários de pico, o que gerava filas na entrada. Usamos essa informação para desenhar o RAV com foco em buscas rápidas e cliques diretos, eliminando a digitação demorada que ocorria no papel.
- **Software Intuitivo e sem Erros:** Os operadores deixaram claro que precisavam de um sistema muito simples, fácil de mexer e que não ficasse dando erros de sistema ou travamentos por falta de internet. Por isso, decidimos configurar o RAV para rodar em um Servidor Local (na própria máquina da guarita) e colocamos travas visuais e máscaras de campos (como na placa e no documento), impedindo que o usuário digite dados errados.
- **Portaria terceirizada (Divisão de Chaves):** Um ponto crucial descoberto nas entrevistas é que a empresa que cuida da portaria é terceirizada, o que significa que os funcionários da guarita mudam constantemente e não

possuem vínculo direto com a secretaria da escola. Essa foi a nossa principal justificativa para criar apenas duas chaves institucionais:

- **Chave de Portaria:** Fica com a empresa terceirizada de plantão registrando entrada e saída, cadastros, controle de estacionamento e geração de relatórios.
- **Chave Administrativa:** Fica exclusiva com a diretoria/gestão da Etec/Fatec, permitindo controle de credenciais e configurações avançadas com segurança.

## 12. FERRAMENTAS E LINGUAGENS UTILIZADAS

Para o desenvolvimento do sistema RAV ETEC — Sistema de Registro de Acesso de Veículos —, foram empregadas tecnologias amplamente consolidadas no mercado de desenvolvimento web, escolhidas com base em sua acessibilidade, documentação abrangente e adequação ao escopo do projeto. A seguir, são descritas as linguagens, bibliotecas, frameworks, APIs externas e ferramentas de ambiente utilizados.

### 12.1 LINGUAGENS DE PROGRAMAÇÃO

#### PHP (versão 8.x)

Principal linguagem de back-end do sistema. Responsável por toda a lógica de negócio do servidor, incluindo autenticação de usuários, processamento de formulários, manipulação de sessões (`session_start()`), consultas ao banco de dados e geração dinâmica de páginas HTML. Funções nativas como `password_hash()` e `password_verify()` foram utilizadas para armazenamento e verificação segura de senhas com o algoritmo **bcrypt** (`PASSWORD_BCRYPT`).

#### HTML5

Linguagem de marcação utilizada para estruturar todas as páginas do sistema. O projeto adota as boas práticas do HTML5, como uso de elementos semânticos (<header>, <nav>, <main>, <footer>, <section>), atributos de acessibilidade e o atributo Lang="pt-BR" para correta indexação e leitura por leitores de tela.

### CSS3

Linguagem de estilos responsável pela identidade visual do sistema. Utilizada em conjunto com o Bootstrap, o CSS3 nativo foi aplicado para customizações específicas de componentes, variáveis CSS (--cps-red, --font-scale), dark mode via classes e animações de transição (efeitos hover, transform, transition). Os arquivos de estilo do projeto incluem: index.css, style.css, painel-controle.css, homepage.css, cadastro.css e login.css.

### JavaScript (ES6+)

Linguagem de programação client-side responsável por toda a interatividade do sistema no navegador. Utilizada para: manipulação do DOM, validações em tempo real de formulários, buscas dinâmicas via fetch() (AJAX), controle de tema (dark/light mode), persistência de preferências via localStorage, exibição de alertas e controle de máscaras de entrada.

### SQL (MySQL Dialect)

Linguagem de consulta estruturada utilizada para criar, manipular e consultar o banco de dados relacional do sistema. O arquivo rav.sql contém o script completo de criação das tabelas, relacionamentos (chaves estrangeiras), índices de performance e dados de exemplo para testes.

## 12.2 BANCO DE DADOS

## MySQL (via MariaDB no Laragon)

Sistema de gerenciamento de banco de dados relacional (SGBD) utilizado para armazenar todos os dados do sistema. O banco de dados possui **12 tabelas principais**, entre elas:

| Tabela           | Finalidade                                              |
|------------------|---------------------------------------------------------|
| unidades         | Armazena dados das ETECs cadastradas e suas credenciais |
| enderecos        | Endereços das unidades (relacionamento 1:1)             |
| condutores       | Pessoas autorizadas a acessar o pátio com veículo       |
| veiculos         | Veículos cadastrados e vinculados a condutores          |
| registros_acesso | Histórico operacional de entradas e saídas              |
| acessos          | Tabela de fluxo de acesso (entrada/saída)               |
| documentos       | Documentos (CPF, CNPJ) dos condutores                   |
| contatos         | Telefones e e-mails dos condutores                      |
| cursos           | Cursos técnicos disponíveis na ETEC                     |
| periodos         | Períodos letivos (Matutino, Vespertino, Noturno)        |
| funcoes          | Funções/cargos dos condutores                           |
| fale_conosco     | Mensagens do formulário de contato público              |

## 12.3 FRAMEWORKS E BIBLIOTECAS FRONT-END

### Bootstrap 5.3.0

Framework CSS e JavaScript mais amplamente utilizado para desenvolvimento de interfaces responsivas. No RAV ETEC, o Bootstrap é responsável pelo sistema de grid, componentes visuais (cards, modals, navbars, formulários, badges, dropdowns, alerts) e utilitários de espaçamento e alinhamento.

Carregado via CDN:

<https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/>

### Bootstrap Icons 1.11.0

Biblioteca de ícones vetoriais (SVG/font) oficial do Bootstrap. Utilizada extensivamente em toda a interface para representar ações, categorias de veículos, status de acesso, menus de navegação e elementos visuais interativos. Mais de 80 ícones distintos são empregados no projeto. Carregada via CDN:

<https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.0/>

### SweetAlert2 (versão 11)

Biblioteca JavaScript para exibição de alertas e diálogos modais estilizados. Substitui os alertas padrão do navegador (alert(), confirm()) por modais com animações, ícones e botões personalizáveis. Utilizada nas páginas de **Painel Admin** e **Acessos** para confirmações de ações críticas e notificações de resultado. Carregada via CDN: <https://cdn.jsdelivr.net/npm/sweetalert2@11>

## IMask.js

Biblioteca JavaScript de mascaramento de campos de entrada. Aplica automaticamente formatações como:

- **CPF:** 000.000.000-00
- **CNPJ:** 00.000.000/0000-00
- **Telefone/Celular:** (00) 0000-0000 / (00) 00000-0000
- **CEP:** 00000-000
- **Placa de veículo:** formato Mercosul/antigo

Carregada via CDN: <https://unpkg.com/imask>

## Google Fonts — Inter

Tipografia principal do sistema. A fonte **Inter** (pesos 300, 400, 500, 600, 700 e 800) foi escolhida por sua legibilidade em telas e identidade visual moderna, alinhada ao padrão institucional do Centro Paula Souza. Carregada via: <https://fonts.googleapis.com/css2?family=Inter>

## 12.4 APIS EXTERNAS

### API ViaCEP

API pública e gratuita utilizada no formulário de cadastro de unidades para preenchimento automático de endereço a partir do CEP digitado. A requisição é feita via `fetch()` em JavaScript: <https://viacep.com.br/ws/{CEP}/json/>

## 12.5 AMBIENTE DE DESENVOLVIMENTO

### Laragon (Windows)

Ambiente de desenvolvimento local para Windows que integra Apache, MySQL (MariaDB), PHP e ferramentas auxiliares em um único pacote portátil. Utilizado para executar o servidor web e o banco de dados localmente durante todo o desenvolvimento do sistema.

### Apache HTTP Server

Servidor web embutido no Laragon, responsável por processar as requisições HTTP e servir os arquivos PHP ao navegador.

### VS Code (Visual Studio Code)

Editor de código-fonte principal utilizado no desenvolvimento. Conta com suporte a PHP, HTML, CSS, JavaScript e SQL, além de extensões para realce de sintaxe, autocompletar e integração com controle de versão.

### Git / GitHub

Sistema de controle de versão utilizado para gerenciar o histórico de alterações do código-fonte e backup do projeto. O GitHub foi utilizado como repositório remoto.

### Navegadores Web (Chrome / Edge)

Utilizados para testes e validação da interface do sistema, incluindo uso das ferramentas de desenvolvedor (DevTools) para inspeção de elementos, depuração de JavaScript e análise de requisições de rede.

## 12.6 RESUMO DAS TECNOLOGIAS

| Categoria              | Tecnologia / Ferramenta                  |
|------------------------|------------------------------------------|
| Back-end               | PHP 8.x                                  |
| Front-end              | HTML5, CSS3, JavaScript (ES6+)           |
| Banco de Dados         | MySQL / MariaDB                          |
| Framework CSS/JS       | Bootstrap 5.3.0                          |
| Ícones                 | Bootstrap Icons 1.11.0                   |
| Alertas modais         | SweetAlert2 v11                          |
| Mascaramento de campos | IMask.js                                 |
| Tipografia             | Google Fonts (Inter)                     |
| API de CEP             | ViaCEP                                   |
| Ambiente local         | Laragon (Apache + MySQL + PHP)           |
| Editor de código       | Visual Studio Code                       |
| Versionamento          | Git / GitHub                             |
| Segurança (senhas)     | Bcrypt via password_hash() nativo do PHP |
| Segurança (consultas)  | PDO com Prepared Statements              |
| Segurança (saída)      | htmlspecialchars() para prevenção de XSS |

### 13. CRONOGRAMA DO TRABALHO

| CRONOGRAMA                 |             |         |         |         |         |     |         |         |             |         |         |               |
|----------------------------|-------------|---------|---------|---------|---------|-----|---------|---------|-------------|---------|---------|---------------|
| Ação                       | A<br>G<br>O | SE<br>T | OU<br>T | NO<br>V | DE<br>Z | JAN | FE<br>V | MA<br>R | A<br>B<br>R | MA<br>I | JU<br>N | Colaboradores |
| Criação da ideia           | ✓           |         |         |         |         |     |         |         |             |         |         | Todos         |
| Pesquisa bibliográfica     |             | ✓       |         |         |         |     |         |         |             |         |         | Todos         |
| Pesquisa de campo          |             | ✓       |         |         |         |     |         |         |             |         |         | Todos         |
| Levantamento de requisitos |             | ✓       |         |         |         |     |         |         |             |         |         | Todos         |
| Mapeamento do sistema      |             |         | ✓       |         |         |     |         |         |             |         |         | Todos         |
| Prototipagem / Layouts     |             |         |         |         | ✓       |     |         |         |             |         |         | Todos         |
| Escolha de tecnologias     |             |         |         |         | ✓       |     |         |         |             |         |         | Lucas e Pedro |
| Desenvolvimento            |             |         |         |         |         | ✓   |         |         |             |         |         | Todos         |
| Documentação               |             |         |         |         |         |     | ✓       | ✓       | ✓           | ✓       |         | Todos         |
| Implementações             |             |         |         |         |         |     |         | ✓       | ✓           | ✓       |         | Todos         |
| Testes / Correções         |             |         |         |         |         |     |         |         |             | ✓       | ✓       | Todos         |
| Finalização                |             |         |         |         |         |     |         |         |             | ✓       | ✓       | Todos         |
| Apresentação               |             |         |         |         |         |     |         |         |             |         | ✓       | Todos         |

### 14. DOCUMENTAÇÃO DO SOFTWARE

A documentação técnica do RAV ETEC — Registro de Acesso de Veículos — apresenta o mapeamento estrutural e lógico da plataforma, descrevendo como o sistema foi arquitetado para atender à rotina operacional das ETECs. O projeto é uma aplicação web executada em servidor local (Laragon/Apache), desenvolvida integralmente em PHP, MySQL, HTML5, CSS3 e JavaScript, sem dependência

de frameworks de back-end externos, o que garante leveza e facilidade de manutenção por equipes técnicas das próprias unidades escolares.

#### 14.1 FLUXOGRAMA DO SISTEMA

O fluxograma a seguir descreve o caminho percorrido pelo usuário dentro do sistema, desde o acesso à página pública até o encerramento da sessão. O fluxo foi desenhado de forma linear e direta, priorizando a agilidade exigida pelos operadores de portaria.

1. Início — Página Pública (index.php) O usuário acessa a página institucional do RAV ETEC, que apresenta informações sobre o sistema, um vídeo demonstrativo e links para o cadastro de novas unidades e para o login.

2. Cadastro de Unidade (cadastro.php). Caso a unidade ainda não esteja registrada, o gestor realiza o cadastro institucional informando os dados da ETEC, dados do responsável e definindo as senhas. O CEP é preenchido automaticamente via integração com a API pública ViaCEP. Ao final, o sistema grava as senhas como hashes bcrypt no banco de dados.

3. Autenticação (login.php → login\_process.php) O usuário insere o código identificador da unidade e a senha. O arquivo login\_process.php busca a unidade no banco pelo `codigo_identificador` e realiza dois testes em sequência via `password_verify()`:

- Desvio A — Senha Administrativa: Se a senha corresponder ao hash da coluna `senha_admin`, a sessão é iniciada com `$_SESSION['acesso'] = 'admin'` e o usuário é redirecionado para `painel-admin.php`.
- Desvio B — Senha de Portaria: Se a senha corresponder ao hash da coluna `senha_portaria`, a sessão é iniciada com `$_SESSION['acesso'] = 'portaria'` e o usuário é redirecionado diretamente para `estacionamento.php`.

- Desvio C — Senha Inválida: Caso nenhum hash coincida, a sessão não é criada e o usuário é retornado para login.php com mensagem de erro.

4a. Fluxo do Operador de Portaria (estacionamento.php) O operador visualiza em tempo real o grid de veículos presentes no pátio, com simulação gráfica de placa Mercosul. Pode registrar novas entradas pelo painel (painel-admin.php → formulário de acesso) e confirmar saídas clicando em cada card de veículo. A tela se atualiza automaticamente a cada 10 segundos via setInterval(), desde que o operador não esteja realizando uma busca ativa.

4b. Fluxo do Gestor Administrativo (painel-admin.php) O arquivo trava\_seguranca.php é incluído no topo de todas as páginas administrativas, verificando se \$\_SESSION['acesso'] === 'admin'. Caso contrário, o acesso é bloqueado e o usuário é redirecionado para estacionamento.php. O gestor tem acesso ao painel completo com as seguintes seções:

| Página                  | Funcionalidade                                               |
|-------------------------|--------------------------------------------------------------|
| painel-admin.php        | Registro de entradas/saídas, busca global por placa ou nome  |
| acessos.php             | Listagem e edição de acessos diários                         |
| estacionamento.php      | Visualização em tempo real do pátio                          |
| gerenciar_cadastros.php | Gerenciamento de condutores e veículos cadastrados           |
| relatorios.php          | Ata de movimentação com filtros por data e gráfico mensal    |
| configuracoes.php       | Dados da unidade, preferências visuais e alteração de senhas |

5. Lógica de Registro de Acesso (registrar\_acesso.php). Ao submeter o formulário de entrada, uma requisição POST é enviada ao servidor. O arquivo PHP realiza a sanitização da placa (strtoupper + preg\_replace), verifica duplicidade (se o veículo já está com status 'Dentro' no banco) e, dentro de uma transação PDO (beginTransaction / commit / rollBack), realiza o upsert do veículo na tabela veiculos e insere o registro na tabela registros\_acesso com status 'Dentro'. A resposta é devolvida em JSON para o front-end.

6. Encerramento de Sessão (sair.php) O arquivo executa session\_unset() e session\_destroy(), limpando todas as variáveis de sessão da memória do servidor, e redireciona o usuário para index.php.

## 14.2 DIAGRAMA DE CASO DE USO

O Diagrama de Caso de Uso descreve as interações de cada ator com as funcionalidades disponíveis no RAV ETEC. O sistema opera com dois perfis de acesso distintos, diferenciados no momento da autenticação por senhas separadas armazenadas na mesma tabela (unidades).

Ator 1 — Operador de Portaria (Senha de Portaria)

Perfil operacional da equipe terceirizada da guarita. Possui acesso restrito às funcionalidades de operação diária:

Visualizar o pátio de veículos em tempo real (estacionamento.php)

Registrar entrada de veículo (formulário com placa, condutor, tipo de acesso e observações)

Consultar veículos cadastrados por CPF ou ID de sete dígitos (busca rápida)

Confirmar a saída de veículos pelo grid do estacionamento

Registrar acessos avulsos sem condutor cadastrado (acesso comum)

Restrição: O nível portaria é bloqueado pelo trava\_seguranca.php ao tentar acessar qualquer página administrativa. O sistema redireciona automaticamente para estacionamento.php?error=acesso\_negado.

## Ator 2 — Gestor Escolar (Senha Administrativa)

Perfil de gestão com acesso irrestrito ao sistema. Além de todas as funcionalidades do Operador, o Gestor pode:

Acessar o painel administrativo completo (painel-admin.php);

Cadastrar, editar e remover condutores e veículos autorizados (gerenciar\_cadastros.php, gerenciar\_condutores.php);

Consultar e editar registros de acesso anteriores (acessos.php, editar\_acesso.php);

Visualizar relatórios e a ata histórica de movimentação com filtros por data, placa e condutor (relatorios.php);

Gerar e imprimir a ata consolidada em PDF por período (imprimir\_relatorio.php)

Alterar dados institucionais, preferências visuais e senhas da unidade (configuracoes.php);

Enviar e receber mensagens pelo módulo de contato (contato.php, processa\_contato.php).

## Ator 3 — Sistema / Banco de Dados (MySQL via PDO)

Ator interno responsável por:

Receber e processar todas as requisições via PDO com Prepared Statements  
Verificar a integridade referencial entre tabelas (chaves estrangeiras com ON DELETE CASCADE);

Validar os hashes criptográficos das senhas via password\_verify() (bcrypt)

Persistir os registros de data e hora precisos (DATETIME, TIMESTAMP) de cada movimentação;

Detectar e bloquear entradas duplicadas (veículo já com status 'Dentro').

## 14.3 MODELO CONCEITUAL DO BANCO DE DADOS

O modelo conceitual descreve as regras de negócio e os relacionamentos entre as entidades do RAV ETEC em alto nível, sem considerar implementação técnica.

Entidades e Relacionamentos:

Uma Unidade (ETEC) possui um ou muitos Veículos (vinculados àquela unidade escolar).

Uma Unidade registra um ou muitos Acessos (histórico de entradas e saídas).

Um Condutor pode possuir um ou mais Veículos (ex.: aluno com carro e moto).

Um Veículo gera um ou muitos Registros de Acesso ao longo do tempo.

Cada Registro de Acesso pertence obrigatoriamente a um único Veículo e armazena o condutor, horário de entrada (`data_hora_entrada`), horário de saída (`data_hora_saida`) e o status atual ('Dentro' ou 'Saiu').

Um Condutor pode ter um Documento (CPF) e um ou mais Contatos (telefone, e-mail ou WhatsApp).

Um Acesso pode ter Dados Acadêmicos opcionais associados (curso, período e função do condutor).

#### **14.4 MODELO LÓGICO DO BANCO DE DADOS**

O modelo lógico descreve a estrutura relacional implementada no banco rav, com os campos, tipos de dados, chaves e relacionamentos utilizados no sistema. Tabela unidades armazena os dados das ETECs cadastradas, credenciais de acesso e dados do gestor.

Tabela enderecos Relacionamento 1:1 com unidades. Armazena o endereço completo preenchido via API ViaCEP.

Tabela condutores Pessoas autorizadas a acessar a ETEC com veículo (cadastro oficial).

Tabela veiculos Veículos vinculados à unidade e opcionalmente a um condutor cadastrado.

Tabela registros\_acesso (tabela operacional principal). Utilizada pelo painel de portaria, relatórios e controle do pátio. Armazena cada evento de entrada e saída em tempo real.

#### **14.5 MODELO FÍSICO DO BANCO DE DADOS**

O modelo físico consiste no script DDL (Data Definition Language) executado diretamente no SGBD MySQL para criar a estrutura otimizada do RAV ETEC. O banco utiliza ENGINE=InnoDB para suporte a transações e integridade referencial, e charset utf8mb4 para suporte completo a caracteres especiais.

## 15. FLUXOGRAMA

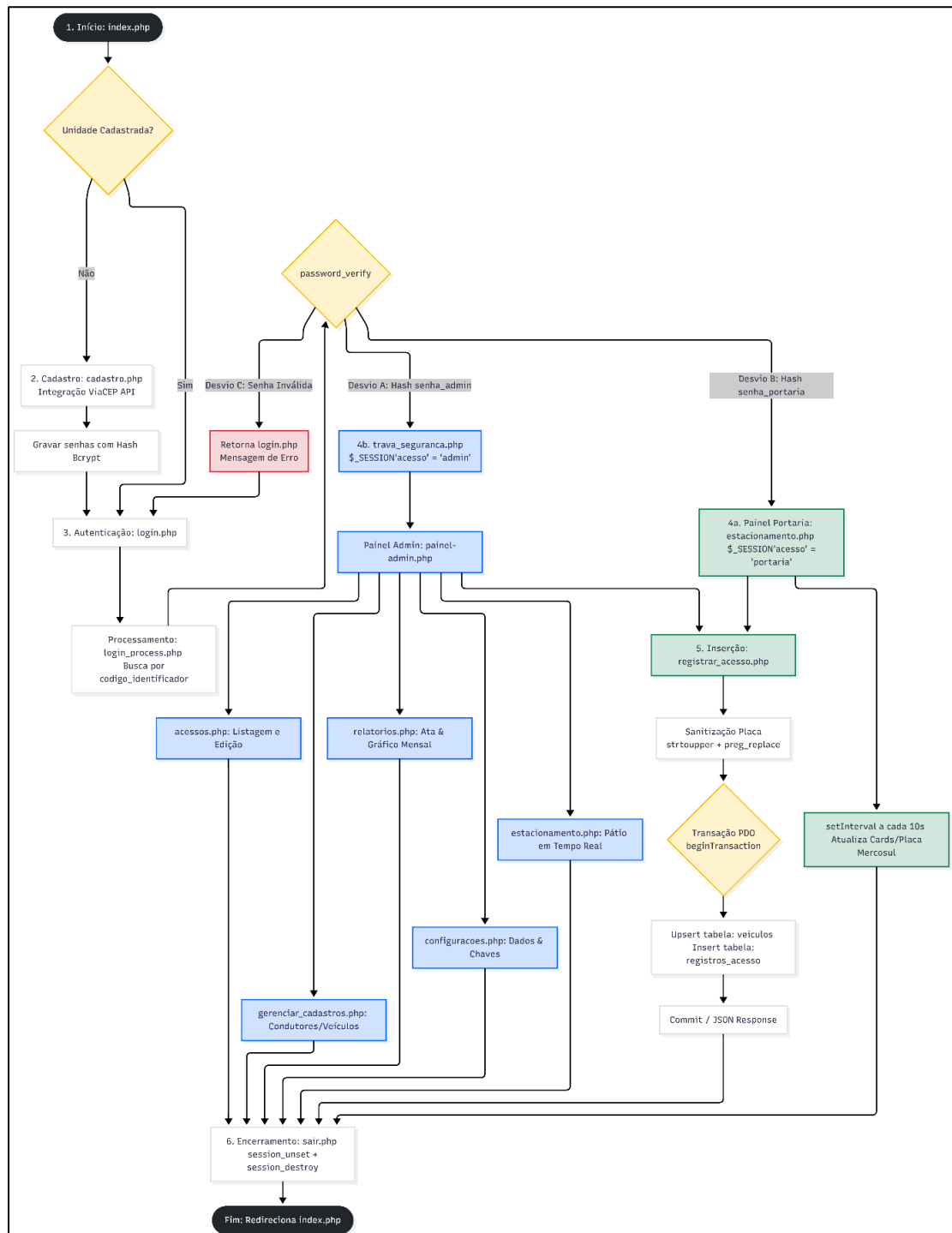


Figura 1 - Fluxograma

## 16. CAPTURA DE TELAS



Figura 2 - Homepage



Figura 3 - Código da homepage

A interface principal do RAV ETEC (index.php) atua como o cartão de visitas institucional da plataforma, apresentando o escopo do projeto, recursos de acessibilidade (ajuste de fontes e alto contraste) e botões de redirecionamento rápido para o login ou registro de novas unidades. O layout foi desenvolvido com foco em responsividade e traz variações nativas para o modo claro (Light Mode) e modo escuro (Dark Mode), garantindo conforto visual e usabilidade em qualquer dispositivo ou iluminação do ambiente.

Figura 4 - Cadastro

```

<!DOCTYPE html>
<html lang="pt-BR" style="--font-scale: 1;"> view-source scroll
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cadastro | RAV ETEC</title>
  <!-- Google Fonts -->
  <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700;800&display=swap" rel="stylesheet">
  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
  <!-- Bootstrap Icons -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.0/font/bootstrap-icons.css">
  <!-- Custom CSS Base do Index -->
  <link rel="stylesheet" href="css/index.css">
</head>
<body class="light-mode d-flex flex-column min-vh-100 section-bg-gray"> flex
  == $0
</html>

```

Figura 5 - Código do cadastro

A interface de cadastro (cadastro.php) foi totalmente reestruturada para unificar as informações em uma única etapa ágil, eliminando a burocracia de criação de perfis individuais. Dividido em dois blocos visuais distintos, o formulário captura os Dados da Unidade (Nome, Código Identificador, Telefone e Endereço) e, simultaneamente, permite definir as Credenciais de Acesso separadas por chaves (senha\_admin e senha\_portaria). O layout limpo, com inputs explicativos e máscaras automáticas, foi projetado para evitar falhas humanas de preenchimento e garantir que a unidade seja configurada com segurança antes de iniciar as operações.

Figura 6 - Login

```

<!DOCTYPE html>
<html lang="pt-BR" style="--font-scale: 1;"> <a href="#">view-source</a> <a href="#">scroll</a>
  <head>...</head>
  <body class="light-mode d-flex flex-column min-vh-100 section-bg-gray"> <a href="#">flex</a>
    <!-- Accessibility Bar (SP Gov & CPS Style) -->
    <div class="accessibility-bar py-1">...</div>
    <!-- Header Foco Total -->
    <header class="main-header bg-white shadow-sm sticky-top border-bottom border-cps-red"
      style="border-bottom: 3px solid var(--cps-red);"> == $0 <a href="#">🔗</a>
      <div class="container py-3 d-flex justify-content-center justify-content-md-between align-items-center">...</div> <a href="#">flex</a>
    </header>
    <!-- Main Content -->
    <main class="container flex-grow-1 d-flex align-items-center justify-content-center py-5">
      ...</main> <a href="#">flex</a>
    <!-- Footer -->
    <footer class="footer-cps bg-dark text-white text-center py-4 mt-auto">...</footer>
    <!-- Bootstrap Bundle -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
    <!-- Scripts de Usabilidade Compartilhados -->
    <script>...</script>
    <style>...</style>
  </body>
</html>

```

Figura 7 - Código do login

A tela de autenticação (login.php) centraliza o acesso à plataforma de forma limpa e intuitiva, exigindo apenas o Código Identificador da unidade de ensino e a respectiva senha. Essa interface é o ponto de partida para a segregação de privilégios controlada pelo back-end. A partir dos dados inseridos, o sistema valida criptograficamente se o usuário possui acesso gerencial restrito ou se será direcionado para o fluxo diário operacional da portaria.

Figura 8 - Painel Operacional

```
<!DOCTYPE html>
<html lang="pt-br" style="--font-scale: 1;"> scroll
<head> == $0
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>RAV Admin</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.0/font/bootstrap-icons.css">
  <link rel="stylesheet" href="css/index.css">
  <link rel="stylesheet" href="css/style.css">
  <style> ... </style>
</head>
<body class="light-mode section-bg-gray d-flex flex-column min-vh-100"> ...
</body> flex
</html>
```

Figura 9 - Código do painel operacional

O Painel Operacional (painel-admin.php / estacionamento.php) é a interface central de trabalho da portaria, projetada especificamente para oferecer máxima agilidade e eliminar o uso de papel. A tela divide-se de forma altamente intuitiva em duas colunas principais:

- Coluna de Registro de Entrada (Esquerda): Permite realizar a busca rápida de condutores já cadastrados por CPF ou ID, agilizando a identificação de alunos

e funcionários. Conta também com a aba 'Acesso Comum' para visitantes avulsos, botões rápidos de seleção de tipo de veículo (Carro, Moto, Outros), classificação do tipo de acesso (Aluno, Equipe, Outros) e um campo de placa com máscara visual automática para evitar erros de digitação.

- Coluna de Registro de Saída (Direita): Exibe uma listagem em tempo real de todos os veículos que estão atualmente no pátio interno da Etec/Fatec, indicando o horário de entrada e o nome do condutor. O operador consegue dar a saída de qualquer veículo com apenas um clique no botão 'Saída', atualizando o status no banco de dados de forma instantânea.

Figura 10 - Cadastro de usuários e automóveis

```
<!DOCTYPE html>
<html lang="pt-br" style="--font-scale: 1;">
  <head>...</head>
  <body class="light-mode section-bg-gray d-flex flex-column min-vh-100 modal-open" style=
    "overflow: hidden; padding-right: 15px;"> flex == $0
    <!-- Acessibility Bar (SP Gov & CPS Style) -->
    <div class="accessibility-bar py-1 d-none d-md-block">...</div>
    <!-- Header Branco (Logo e Usuário) -->
    <header class="main-header bg-white shadow-sm sticky-top">...</header>
    <!-- Navbar Horizontal vermelha padronizada -->
    <nav class="navbar navbar-expand-lg nav-cps p-0" style="z-index: 1010;">...</nav> flex
    <main class="container py-4">...</main>
    <div class="modal fade show" id="modalCadastro" tabindex="-1" aria-modal="true" role=
      "dialog" style="display: block; padding-left: 0px;">...</div> scroll
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.mi
      n.js"></script>
    <script src="https://unpkg.com/imask"></script>
    <script src="script/script.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
    <!-- Scripts de Usabilidade Compartilhados (RAV ETEC) -->
    <script>...</script>
    <div class="modal-backdrop fade show"></div>
  </body>
</html>
```

Figura 11 - Código do cadastro de usuários e automóveis

Esta interface apresenta o componente flutuante (Modal) de Cadastro Completo de Usuários e Automóveis, acessível diretamente pelo painel operacional. Desenvolvido para centralizar o vínculo entre quem dirige e o veículo autorizado, o formulário divide-se em:

- Dados do Condutor: Campos para preenchimento de Nome Completo, E-mail e CPF com aplicação de máscaras automáticas de digitação para garantir a integridade das informações no banco de dados.
- Dados do Veículo: Inputs estruturados para seleção do tipo de veículo, inserção da placa no padrão Mercosul/Nacional, Marca/Modelo e Cor do automóvel.

A gravação desse formulário executa uma transação segura no banco de dados MySQL, permitindo que o operador da guarita terceirizada registre rapidamente um novo aluno ou professor sem precisar sair da tela principal de monitoramento, reduzindo filas na portaria nos horários de pico.

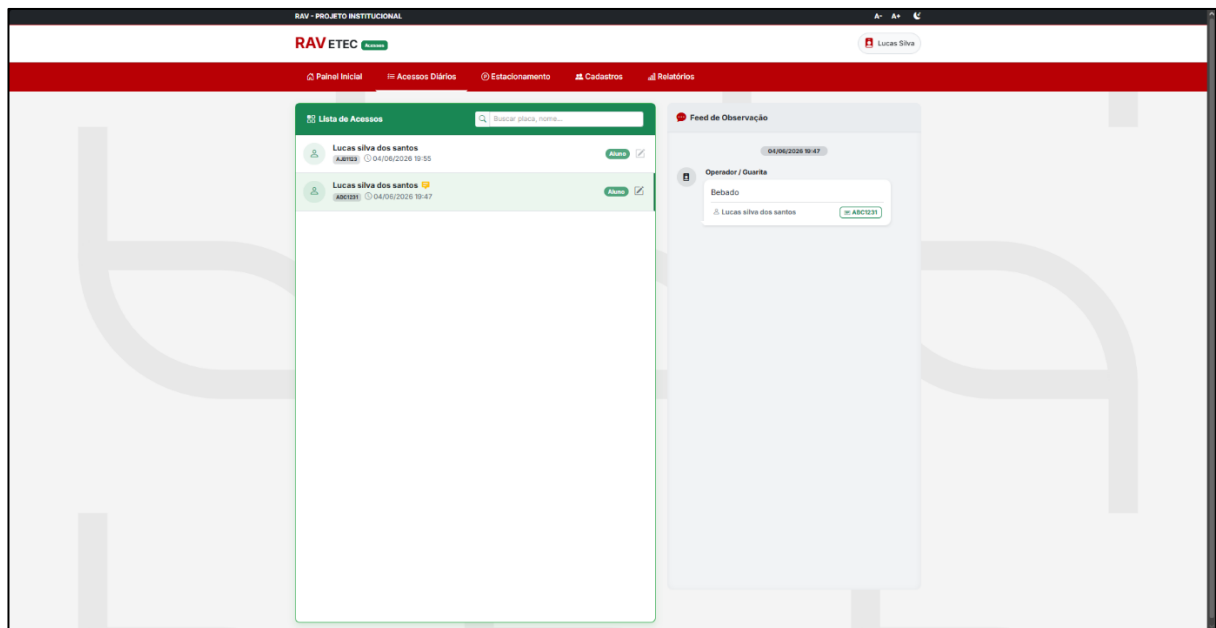


Figura 12 - Acessos

```

<!DOCTYPE html>
<html lang="pt-br"> scroll
  <head> ... </head>
  <body class="light-mode section-bg-gray d-flex flex-column min-vh-100"> flex
    <!-- Navegação / Header Idêntico ao Painel Admin -->
    <div class="accessibility-bar py-1 d-none d-md-block">
      <div class="container d-flex justify-content-between align-items-center"> ... </div>
    </div>
    <header class="main-header bg-white shadow-sm sticky-top">
      <div class="container py-3 d-flex justify-content-between align-items-center"> ... </div>
    </header>
    <nav class="navbar navbar-expand-lg nav-cps p-0" style="z-index: 1010;"> flex == $0
      <div class="container flex-column flex-lg-row"> ... </div>
    </nav>
    <main class="container py-4">
      <div class="row g-4 h-100 align-items-stretch"> ... </div>
    </main>
  </body>
</html>

```

Figura 13 - Código de acessos

A tela de Acessos Diários (acessos.php) atua como uma central de auditoria e monitoramento de ocorrências da portaria. À esquerda, a Lista de Acessos exibe cronologicamente o histórico de movimentações do dia, identificando o condutor, o horário exato e o vínculo institucional (ex: Aluno). À direita, o sistema integra o Feed de Observações, uma funcionalidade estratégica que exibe em tempo real alertas e anotações críticas inseridas pela equipe terceirizada da guarita (como incidentes

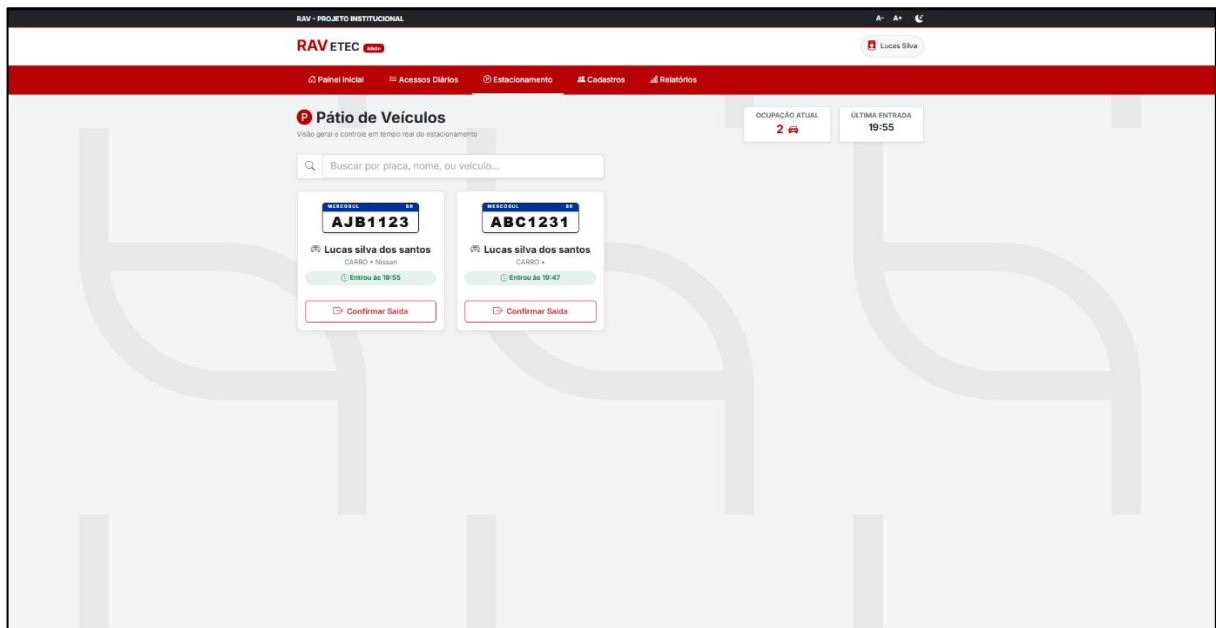


Figura 14- Estacionamento



Figura 15 - Código do estacionamento

A tela de Pátio de Veículos (estacionamento.php) oferece ao operador uma visão geral e controle em tempo real do fluxo interno do estacionamento da unidade. Os veículos presentes no pátio são renderizados graficamente em formato de cards com simulação de placas no padrão Mercosul, exibindo o nome do proprietário, marca/modelo e o carimbo de data e hora de entrada. No topo, indicadores numéricos resumem a Ocupação Atual de vagas e o horário da Última Entrada. Graças ao recurso de atualização automática via JavaScript (setInterval), o grid se mantém sincronizado a cada 10 segundos, permitindo que o porteiro terceirizado confirme saídas rapidamente com um único clique no botão 'Confirmar Saída'.

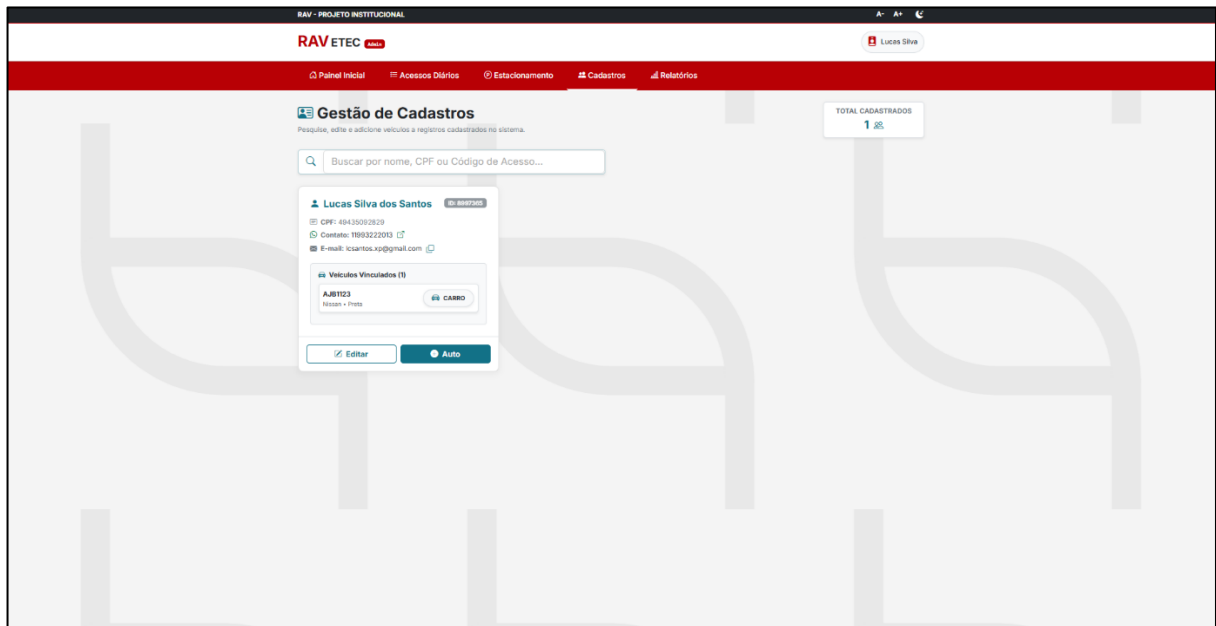


Figura 16 - Gestão de cadastro

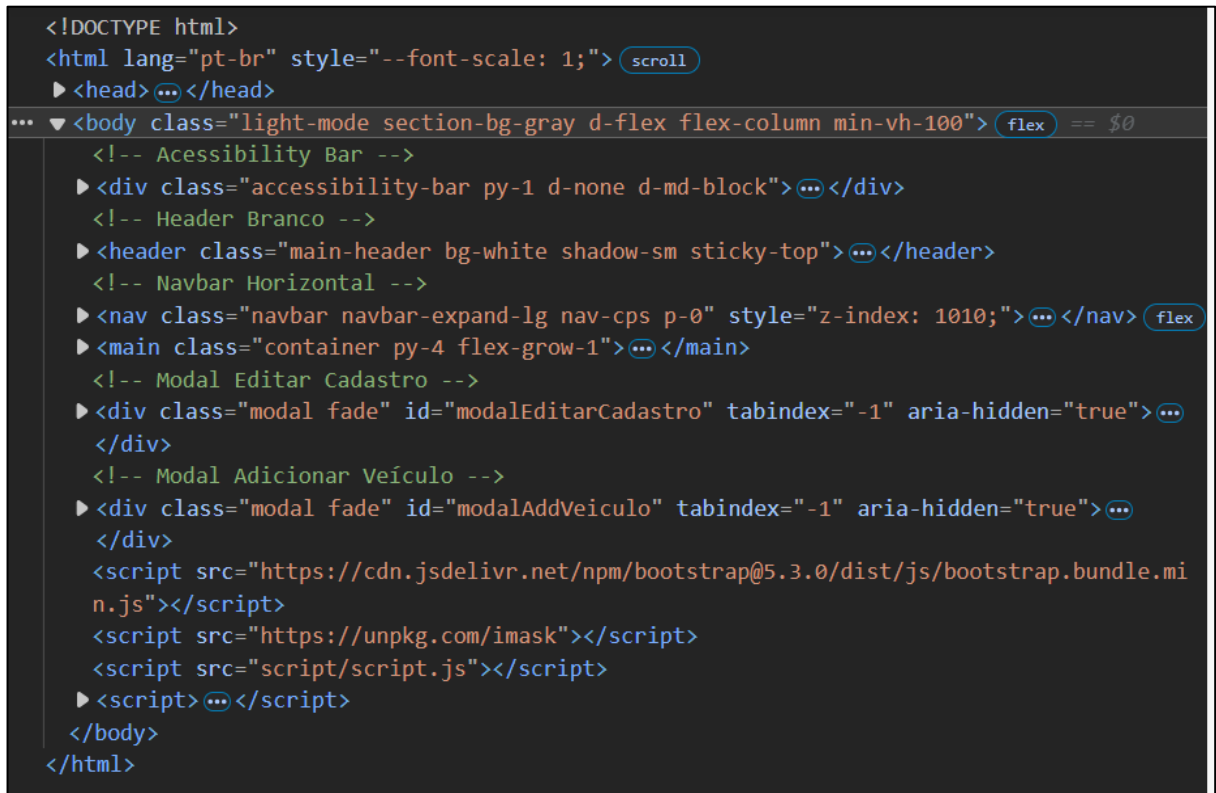


Figura 17 - Código da gestão de cadastro

Esta interface permite a busca global e o gerenciamento centralizado de condutores (professores, alunos e funcionários) e seus respectivos veículos homologados. Ao localizar um registro, o sistema renderiza um card detalhado contendo o nome completo, CPF, dados de contato e a lista de Veículos Vinculados com suas especificações (como placa e modelo). A partir deste painel, o gestor

escolar pode executar rotinas de exclusão ou edição de dados através de Prepared Statements, garantindo a integridade referencial com o banco de dados MySQL.

A tela de Ata de Acessos e Relatórios (relatorios.php) oferece ferramentas avançadas de auditoria e prestação de contas para a direção da Etec/Fatec. No topo, o sistema disponibiliza filtros de busca rápida cruzando dados por placa, condutor e intervalos estritos de Data Inicial e Data Final. O bloco principal apresenta o Histórico de Movimentação (Ata), exibindo de forma clara a data/hora exata, o tipo de movimento (Entrada ou Saída), o nome do condutor e os detalhes do veículo. Assim essas informações podem ser usadas caso o condutor esqueça o veículo na unidade. Adicionalmente, a interface integra o módulo imprimir relatório para a geração e extração consolidada da ata em formato PDF por período, além de um gráfico dinâmico de Frequência de Acesso no mês atual para análise estatística do fluxo perimetral.

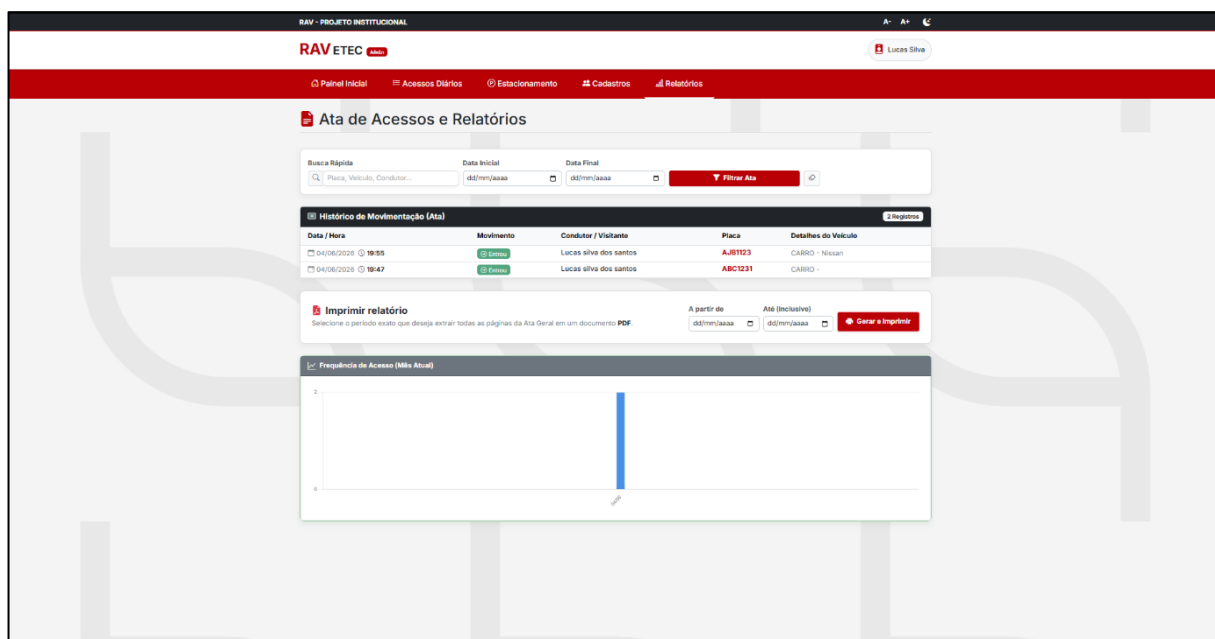


Figura 18 - Relatório

```

<!DOCTYPE html>
<html lang="pt-br" style="--font-scale: 1;"> scroll
<head> == $0
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>RAV Admin - Relatórios</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.0/font/bootstrap-icons.css">
  <link rel="stylesheet" href="css/index.css">
  <link rel="stylesheet" href="css/style.css">
</head>
<body class="light-mode section-bg-gray d-flex flex-column min-vh-100"> flex
</body>
</html>

```

Figura 19 - Código do relatório

Figura 20 - Configurações

```

<!DOCTYPE html>
<html lang="pt-br" style="--font-scale: 1;">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>RAV Admin – Configurações do Sistema</title>
  <meta name="description" content="Painel de configurações seguro da unidade E
  TEC no sistema RAV.">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.m
  in.css" rel="stylesheet">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.1
  1.0/font/bootstrap-icons.css">
  <link rel="stylesheet" href="css/index.css">
  <link rel="stylesheet" href="css/style.css">
  <style>
  </style>
</head>
<body class="light-mode section-bg-gray d-flex flex-column min-vh-100">
</body>
</html>

```

Figura 21 - Código das configurações

A tela de Configurações do Sistema permite ao administrador realizar o gerenciamento das principais configurações da aplicação. Nela é possível atualizar informações da conta, como nome, e-mail, telefone e código de acesso, além de confirmar a senha administrativa para aplicar alterações. Também possui opções de personalização da interface, como modo escuro, animações e ajuste do tamanho da fonte. Na área de alertas e notificações, o usuário pode configurar avisos relacionados ao funcionamento do sistema. A seção de segurança permite alterar senhas de acesso, garantindo maior controle, proteção dos dados e segurança das operações realizadas na plataforma.

## 17. DIAGRAMA DE CASO DE USO

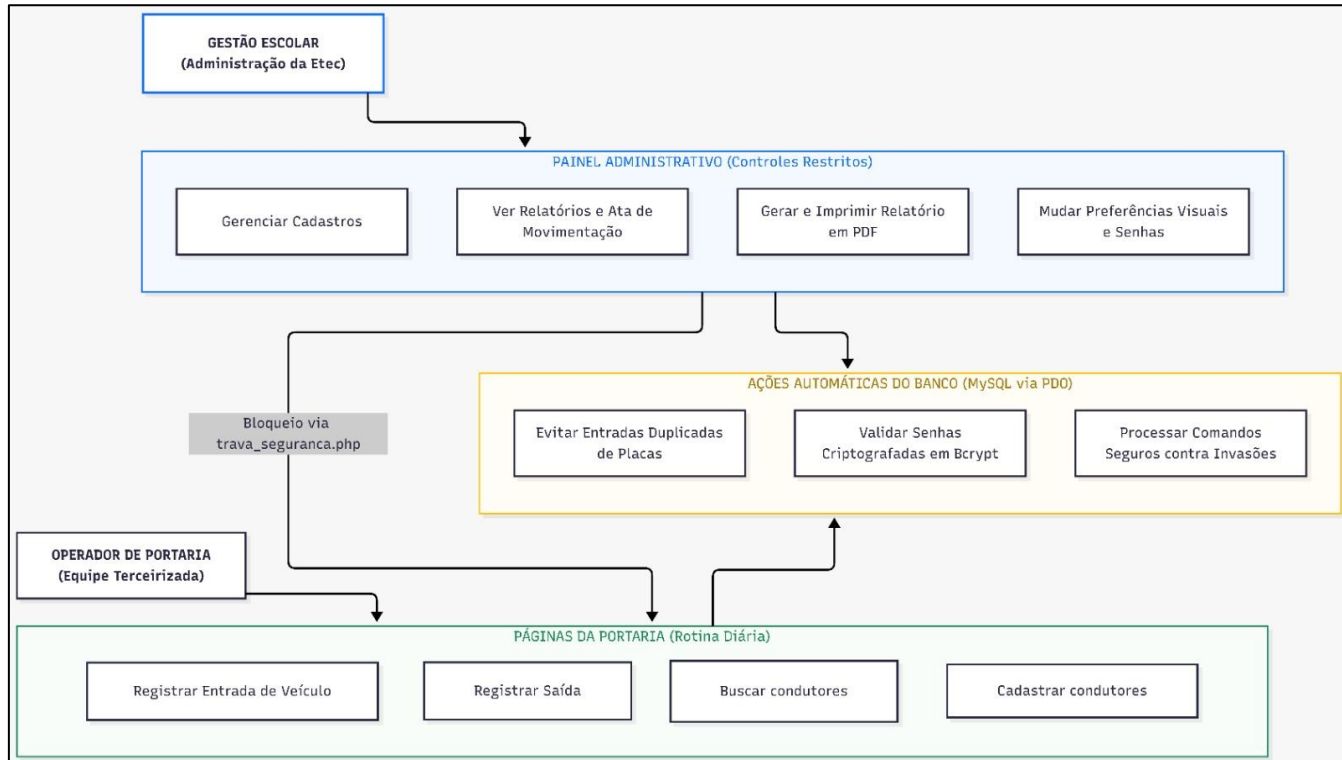


Figura 22 - Diagrama de caso de uso

## 18. MODELO CONCEITUAL DO BANCO DE DADOS

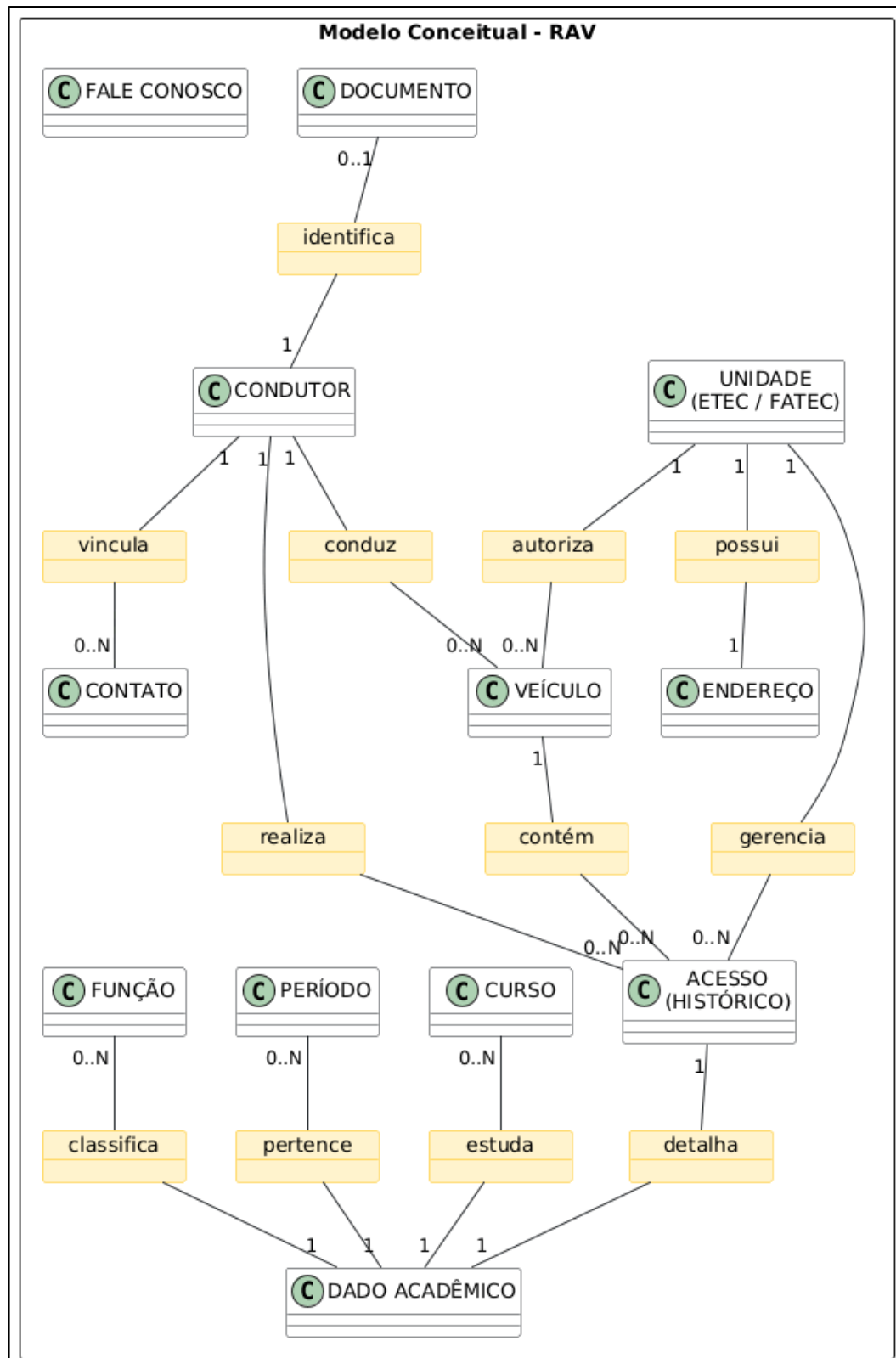


Figura 23 - Diagrama de caso de uso

## 19. MODELO LÓGICO DO BANCO DE DADOS

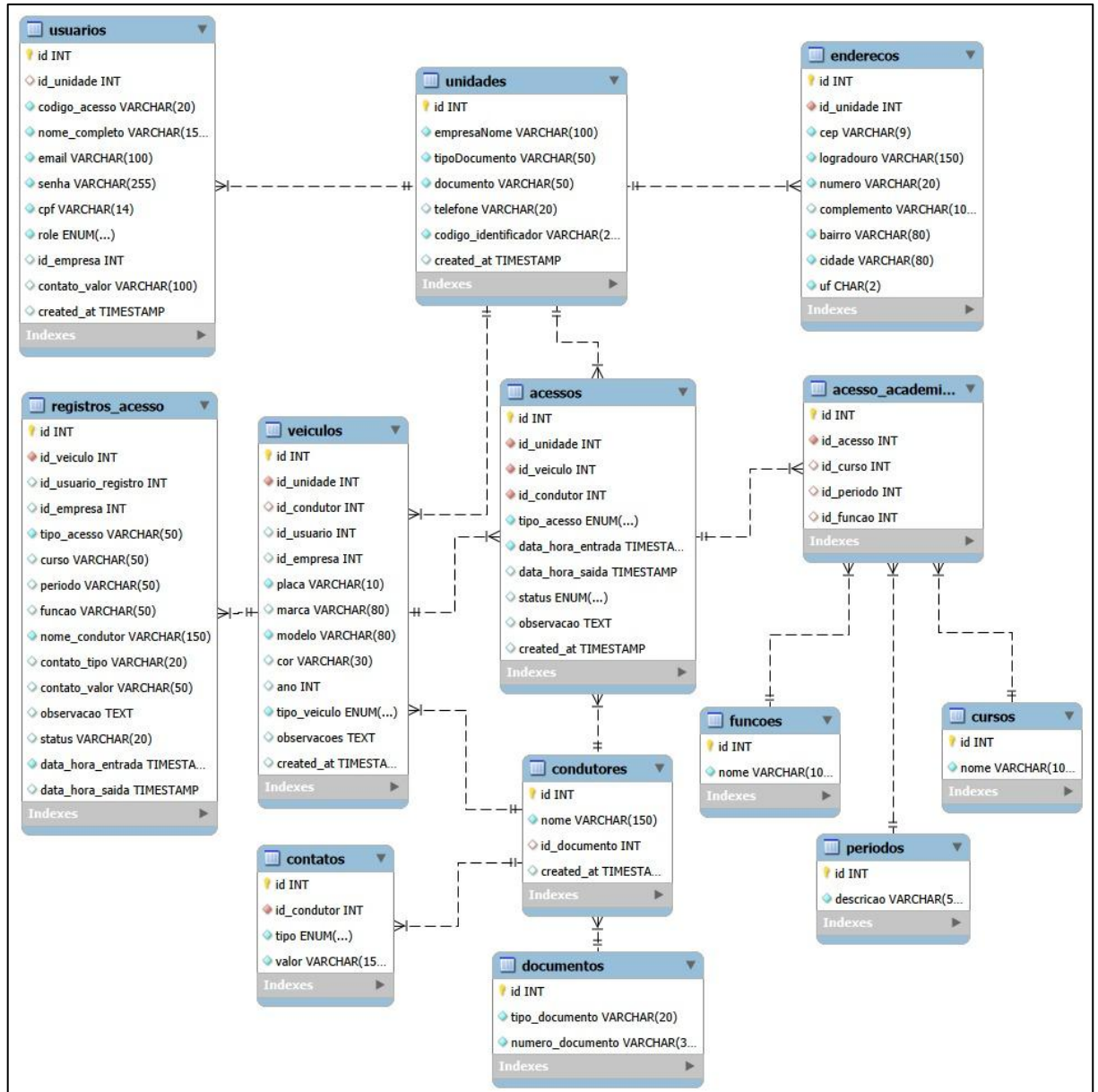


Figura 24 - Modelo lógico do banco de dados

## 20. MODELO FÍSICO DO BANCO DE DADOS

```
CREATE DATABASE IF NOT EXISTS rav CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci;  
USE rav;
```

```
SET FOREIGN_KEY_CHECKS = 0;
```

```
DROP TABLE IF EXISTS registros_acesso;  
DROP TABLE IF EXISTS usuarios;  
DROP TABLE IF EXISTS acesso_academico;  
DROP TABLE IF EXISTS acessos;  
DROP TABLE IF EXISTS veiculos;  
DROP TABLE IF EXISTS contatos;  
DROP TABLE IF EXISTS condutores;  
DROP TABLE IF EXISTS funcoes;  
DROP TABLE IF EXISTS periodos;  
DROP TABLE IF EXISTS cursos;  
DROP TABLE IF EXISTS documentos;  
DROP TABLE IF EXISTS enderecos;  
DROP TABLE IF EXISTS unidades;  
DROP TABLE IF EXISTS fale_conosco;
```

```
CREATE TABLE unidades (  
id INT AUTO_INCREMENT PRIMARY KEY,  
empresaNome VARCHAR(100) NOT NULL,  
tipoDocumento VARCHAR(50) NOT NULL,  
documento VARCHAR(50) NOT NULL UNIQUE,  
telefone VARCHAR(20) NULL,  
nome_completo VARCHAR(100) NOT NULL,  
email VARCHAR(100) NOT NULL,  
cpf VARCHAR(14) NOT NULL,  
datanasc DATE NULL,
```

```
codigo_identificador VARCHAR(20) NOT NULL UNIQUE,  
senha_admin VARCHAR(255) NOT NULL,  
senha_portaria VARCHAR(255) NOT NULL,  
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TABLE enderecos (  
id INT AUTO_INCREMENT PRIMARY KEY,  
id_unidade INT NOT NULL UNIQUE,  
cep VARCHAR(9) NOT NULL,  
logradouro VARCHAR(150) NOT NULL,  
numero VARCHAR(20) NOT NULL,  
complemento VARCHAR(100) NULL,  
bairro VARCHAR(80) NOT NULL,  
cidade VARCHAR(80) NOT NULL,  
uf CHAR(2) NOT NULL,  
CONSTRAINT fk_endereco_unidade  
FOREIGN KEY (id_unidade)  
REFERENCES unidades(id)  
ON DELETE CASCADE  
);
```

```
CREATE TABLE documentos (  
id INT AUTO_INCREMENT PRIMARY KEY,  
tipo_documento VARCHAR(20) NOT NULL,  
numero_documento VARCHAR(30) NOT NULL UNIQUE  
);
```

```
CREATE TABLE cursos (  
id INT AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(100) NOT NULL UNIQUE  
);
```

```
CREATE TABLE periodos (  

```

```
id INT AUTO_INCREMENT PRIMARY KEY,  
descricao VARCHAR(50) NOT NULL UNIQUE  
);
```

```
CREATE TABLE funcoes (  
id INT AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(100) NOT NULL UNIQUE  
);
```

```
CREATE TABLE condutores (  
id INT AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(150) NOT NULL,  
id_documento INT NULL,  
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
CONSTRAINT fk_condutor_documento  
FOREIGN KEY (id_documento)  
REFERENCES documentos(id)  
ON DELETE SET NULL  
);
```

```
CREATE TABLE contatos (  
id INT AUTO_INCREMENT PRIMARY KEY,  
id_condutor INT NOT NULL,  
tipo ENUM('EMAIL', 'TELEFONE', 'WHATSAPP') NOT NULL,  
valor VARCHAR(150) NOT NULL,  
CONSTRAINT fk_contato_condutor  
FOREIGN KEY (id_condutor)  
REFERENCES condutores(id)  
ON DELETE CASCADE  
);
```

```
CREATE TABLE veiculos (  
id INT AUTO_INCREMENT PRIMARY KEY,  
id_unidade INT NOT NULL,
```

```

id_condutor INT NULL,
id_usuario INT NULL,
id_empresa INT NULL,
placa VARCHAR(10) NOT NULL UNIQUE,
marca VARCHAR(80) NULL,
modelo VARCHAR(80) NOT NULL,
cor VARCHAR(30) NULL,
ano INT NULL,
tipo_veiculo ENUM('CARRO', 'MOTO', 'CAMINHAO', 'ONIBUS', 'VAN', 'OUTRO')
NOT NULL,
observacoes TEXT NULL,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
CONSTRAINT fk_veiculo_unidade
FOREIGN KEY (id_unidade)
REFERENCES unidades(id)
ON DELETE CASCADE,
CONSTRAINT fk_veiculo_condutor
FOREIGN KEY (id_condutor)
REFERENCES condutores(id)
ON DELETE SET NULL
);

```

```

CREATE TABLE acessos (
id INT AUTO_INCREMENT PRIMARY KEY,
id_unidade INT NOT NULL,
id_veiculo INT NOT NULL,
id_condutor INT NOT NULL,
tipo_acesso ENUM('ENTRADA', 'SAIDA', 'ENTRADA_SAIDA') NOT NULL,
data_hora_entrada DATETIME NOT NULL,
data_hora_saida DATETIME NULL,
status ENUM('ABERTO', 'EM_ANDAMENTO', 'FINALIZADO', 'CANCELADO')
DEFAULT 'ABERTO',
observacao TEXT NULL,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

```

```

CONSTRAINT fk_acesso_unidade
FOREIGN KEY (id_unidade)
REFERENCES unidades(id)
ON DELETE CASCADE,
CONSTRAINT fk_acesso_veiculo
FOREIGN KEY (id_veiculo)
REFERENCES veiculos(id)
ON DELETE CASCADE,
CONSTRAINT fk_acesso_condutor
FOREIGN KEY (id_condutor)
REFERENCES condutores(id)
ON DELETE CASCADE
);

```

```

CREATE TABLE acesso_academico (
id INT AUTO_INCREMENT PRIMARY KEY,
id_acesso INT NOT NULL,
id_curso INT NULL,
id_periodo INT NULL,
id_funcao INT NULL,
CONSTRAINT fk_academico_acesso
FOREIGN KEY (id_acesso)
REFERENCES acessos(id)
ON DELETE CASCADE,
CONSTRAINT fk_academico_curso
FOREIGN KEY (id_curso)
REFERENCES cursos(id)
ON DELETE SET NULL,
CONSTRAINT fk_academico_periodo
FOREIGN KEY (id_periodo)
REFERENCES periodos(id)
ON DELETE SET NULL,
CONSTRAINT fk_academico_funcao
FOREIGN KEY (id_funcao)

```

```
REFERENCES funcoes(id)
ON DELETE SET NULL
);
```

```
CREATE TABLE fale_conosco (
id INT AUTO_INCREMENT PRIMARY KEY,
nome VARCHAR(100) NOT NULL,
email VARCHAR(100) NOT NULL,
telefone VARCHAR(20) NULL,
assunto VARCHAR(150) NOT NULL,
mensagem TEXT NOT NULL,
data_envio DATETIME DEFAULT CURRENT_TIMESTAMP,
status ENUM('Pendente', 'Lido', 'Respondido') DEFAULT 'Pendente'
);
```

```
CREATE TABLE usuarios (
id INT AUTO_INCREMENT PRIMARY KEY,
codigo_acesso VARCHAR(20) NOT NULL UNIQUE,
nome_completo VARCHAR(150) NOT NULL,
email VARCHAR(100) NOT NULL UNIQUE,
senha VARCHAR(255) NOT NULL,
cpf VARCHAR(14) NOT NULL,
id_empresa INT NULL,
role VARCHAR(50) DEFAULT 'visitante',
contato_valor VARCHAR(100) NULL,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
CREATE TABLE registros_acesso (
id INT AUTO_INCREMENT PRIMARY KEY,
id_veiculo INT NOT NULL,
id_usuario_registro INT NULL,
id_empresa INT NULL,
tipo_acesso VARCHAR(50) NOT NULL,
```

```
curso VARCHAR(50) NULL,  
periodo VARCHAR(50) NULL,  
funcao VARCHAR(50) NULL,  
nome_condutor VARCHAR(150) NOT NULL,  
contato_tipo VARCHAR(20) NULL,  
contato_valor VARCHAR(50) NULL,  
observacao TEXT NULL,  
status VARCHAR(20) DEFAULT 'Dentro',  
data_hora_entrada DATETIME DEFAULT CURRENT_TIMESTAMP,  
data_hora_saida DATETIME NULL,  
CONSTRAINT fk_registro_veiculo  
FOREIGN KEY (id_veiculo)  
REFERENCES veiculos(id)  
ON DELETE CASCADE  
);  
  
CREATE INDEX idx_veiculo_unidade ON veiculos(id_unidade);  
CREATE INDEX idx_veiculo_placa ON veiculos(placa);  
CREATE INDEX idx_acesso_entrada ON acessos(data_hora_entrada);  
CREATE INDEX idx_acesso_status ON acessos(status);  
CREATE INDEX idx_condutor_nome ON condutores(nome);  
  
SET FOREIGN_KEY_CHECKS = 1;
```

## **21. CICLO DE VIDA DO PROGRAMA**

O ciclo de vida adotado para o desenvolvimento do RAV (Registro de Acesso de Veículos) baseia-se no Modelo Incremental e Evolutivo, orientado pelas práticas da metodologia ágil Scrum. Diferente dos modelos tradicionais (como o modelo em Cascata, onde uma fase só começa quando a outra termina), o modelo incremental permitiu que o sistema fosse construído, testado e refinado em ciclos repetitivos chamados Sprints, garantindo a entrega de um software funcional dentro do prazo acadêmico.

### **1. PLANEJAMENTO E VISÃO DO PRODUTO (PRODUCT BACKLOG)**

O ciclo iniciou-se com a transformação das reclamações coletadas nas portarias terceirizadas em metas de desenvolvimento. Em vez de criarmos um sistema complexo de usuários individuais, o planejamento focou na simplificação do escopo: a criação das duas chaves institucionais (Administração e Portaria). Todas as funcionalidades desejadas foram listadas e priorizadas por ordem de importância para a segurança da Etec/Fatec.

### **2. ANÁLISE E DESIGN (SPRINT PLANNING & PROTOTIPAGEM)**

A cada ciclo de desenvolvimento, o grupo selecionava quais telas e funções seriam programadas. Nesta fase, foram desenhados os diagramas de caso de uso, o fluxo do usuário (index.php  $\rightarrow$  login.php  $\rightarrow$  desvios de chaves) e a modelagem do banco de dados MySQL. O design visual foi planejado usando o Bootstrap 5 para garantir que a interface fosse totalmente intuitiva e rápida para o operador da guarita.

### **3. IMPLEMENTAÇÃO E CODIFICAÇÃO (SPRINT)**

Nesta fase, as ideias foram transformadas em código real utilizando PHP nativo, JavaScript e o banco de dados. Os programadores focaram na construção de uma arquitetura limpa:

Desenvolvimento do back-end com Prepared Statements via PDO para blindar o sistema contra invasões.

Criação das travas de segurança (trava\_seguranca.php) para impedir que a portaria terceirizada acessasse dados administrativos.

#### **4. TESTES E INTEGRAÇÃO (SPRINT REVIEW)**

Testes de Autenticação: Validação se a função `password_verify()` estava quebrando o loop de login e identificando corretamente os hashes criptografados em Bcrypt.

Testes de Input: Verificação se as máscaras do IMask estavam limpando corretamente os pontos e traços de documentos (CPF/CNPJ) antes de salvar no banco.

## **22. CONSIDERAÇÕES FINAIS**

O desenvolvimento do sistema RAV (Registro de Acesso de Veículos) cumpriu com êxito os objetivos propostos ao consolidar uma ferramenta prática para a gestão de fluxo no perímetro das Etecs e FATECs. A transição do modelo tradicional de pranchetas em papel para uma plataforma web automatizada provou-se viável, eficiente e totalmente alinhada com as necessidades reais levantadas na pesquisa de campo.

A decisão de arquitetura focada em apenas duas chaves institucionais unificadas (Administrativa e Portaria) mostrou-se o maior acerto estratégico do projeto, solucionando de forma simples o problema da alta rotatividade e falta de vínculo direto das equipes de portaria terceirizadas

Como propostas para trabalhos futuros e continuidade do RAV, sugere-se a expansão da plataforma para integração nativa com barreiras físicas (como catracas e cancelas automatizadas) e o desenvolvimento de um módulo de inteligência para leitura automática de placas via câmera (OCR), o que elevaria ainda mais o nível de automação. Em suma buscamos uma solução realista, de baixo custo e pronta para ser implantada nas unidades do Centro Paula Souza.

## 23. REFERÊNCIAS

- PIMENTEL, Arthur Moraes. **Protótipo para controle de acesso veicular utilizando tecnologia RFID**. 2023. Disponível em: [https://www.eng-computacao.timoteo.cefetmg.br/wp-content/uploads/sites/197/2025/01/TCC\\_125.pdf](https://www.eng-computacao.timoteo.cefetmg.br/wp-content/uploads/sites/197/2025/01/TCC_125.pdf). Acesso em: 10 jul. 2026.
- FERRO, Júlio César; BAMPA, Rubens. **Sistema web de gerenciamento de estacionamentos para o controle de fluxo de veículos**. 2021. Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) – Campus Barretos. Disponível em: <https://brt.ifsp.edu.br/phocadownload/userupload/213354/IFMIN220001.pdf>. Acesso em: 10 jul. 2026.
- SOUZA, Marcos Tork. **Controle de acesso para sistemas distribuídos**. 2010. Universidade de São Paulo (USP). Disponível em: <https://www.teses.usp.br/teses/disponiveis/3/3141/tde-23052011-111816/pt-br.php>. Acesso em: 10 jul. 2026.