

Manual de Uso - Sistema de Quilometragem Veicular

Visão Geral

Sistema completo para registro e controle de quilometragem de veículos, desenvolvido com NextJS 14, TypeScript, PostgreSQL e Prisma ORM.

Início Rápido

Instalação Local

```
# 1. Clone ou navegue até o diretório
cd /home/ubuntu/vehicle_mileage_system

# 2. Instale as dependências
npm install

# 3. Configure o arquivo .env com suas credenciais PostgreSQL
# Edite o arquivo .env com a URL do seu banco de dados

# 4. Execute as migrações do banco
npx prisma migrate dev

# 5. Popule o banco com dados iniciais
npm run prisma:seed

# 6. Inicie o servidor de desenvolvimento
npm run dev
```

Acesse: <http://localhost:3000>

Credenciais Padrão

Após executar o seed:

- **Email:** admin@quilometragem.com
- **Senha:** admin123

 **Altere a senha após o primeiro acesso!**

Funcionalidades

1. Autenticação

Login

- Acesse `/login`
- Digite email e senha
- Clique em “Entrar”

Registro

- Acesse `/register`

- Preencha nome, email e senha
- Clique em “Registrar”
- Faça login com as credenciais criadas

2. Dashboard Principal

Visualizar Registros

- Após login, você verá a lista de todos os registros
- Informações exibidas:
 - Condutor
 - Placa do veículo
 - Data/hora de saída e chegada
 - Distância percorrida


Criar Novo Registro

1. Clique em “Novo Registro”
2. Preencha o formulário:
 - Selecione o veículo
 - Digite o nome do condutor
 - Data/hora de saída
 - Data/hora de chegada
 - KM inicial e final
 - Motivo (opcional)
 - Observações (opcional)
3. Clique em “Salvar”

Validações:


- KM final deve ser maior que KM inicial
- Todos os campos obrigatórios devem ser preenchidos

Editar Registro


1. Clique no ícone de lápis () ao lado do registro
2. Modifique os campos desejados
3. Clique em “Atualizar”

Nota: Todas as alterações são registradas no histórico!

Ver Histórico de Alterações

1. Clique no ícone de histórico () ao lado do registro
2. Visualize todas as modificações:
 - Campo alterado
 - Valor anterior
 - Valor novo
 - Usuário que fez a alteração
 - Data/hora da alteração

Deletar Registro

1. Clique no ícone de lixeira () ao lado do registro
2. Confirme a exclusão

 **Esta ação não pode ser desfeita!**

3. Exportações

Exportar CSV

- Clique em “Exportar CSV”
- Arquivo será baixado automaticamente
- Formato: quilometragem.csv
- Pode ser aberto no Excel, Google Sheets, etc.

Exportar Excel

- Clique em “Exportar Excel”
- Arquivo será baixado automaticamente
- Formato: quilometragem.xlsx
- Arquivo nativo do Microsoft Excel

Exportar PDF

- Clique em “Exportar PDF”
- Arquivo será baixado automaticamente
- Formato: quilometragem.pdf
- Pronto para impressão

Filtros de Exportação:

Todas as exportações podem ser filtradas por:

- Data de início
- Data de fim

4. QR Code

Gerar QR Code

1. Clique em “QR Code” no header
2. QR Code será exibido em nova janela
3. Escaneie com smartphone para acesso rápido
4. Salve a imagem se necessário

Uso prático:

- Cole o QR Code em áreas visíveis
- Motoristas podem escanear para acessar rapidamente
- Facilita o registro no momento da saída/chegada

5. Gerenciamento de Veículos

Para adicionar novos veículos, você pode:

Via Interface (Planejado)

Em desenvolvimento

Via API

```
curl -X POST http://localhost:3000/api/vehicles \
  -H "Content-Type: application/json" \
  -d '{
    "licensePlate": "XYZ-5678",
    "brand": "Honda",
    "model": "Civic",
    "year": 2024,
    "color": "Preto"
  }'
```

Via Prisma Studio

```
npx prisma studio
# Acesse http://localhost:5555
# Adicione veículos pela interface visual
```



Configuração

Modo de Produção vs Modo de Teste

Modo de Produção (Padrão)

```
APP_MODE=production
DATABASE_URL=postgresql://...
```

- Requer autenticação
- Registros associados a usuários
- Histórico completo de alterações

Modo de Teste

```
APP_MODE=test
DATABASE_URL_TEST=postgresql://...
```

- Sem autenticação necessária
- Qualquer pessoa pode adicionar/editar registros
- Ideal para demonstrações e testes
- Banco de dados separado

Banco de Dados

PostgreSQL Local

```
DATABASE_URL="postgresql://user:password@localhost:5432/vehicle_mileage"
```

PostgreSQL em Cloud (Neon, Supabase, etc.)

```
DATABASE_URL="postgresql://user:password@host.cloud.com:5432/database?sslmode=require"
```

Interface

Responsividade

- ☒ Desktop (1920x1080 e superior)
- ☒ Laptop (1366x768)
- ☒ Tablet (768x1024)
- ☒ Mobile (360x640 e superior)

Navegação Mobile

- Menu hambúrguer (planejado)
- Tabelas com scroll horizontal
- Formulários em tela cheia
- Botões otimizados para toque

Segurança

Autenticação

- NextAuth.js para gerenciamento de sessões
- Senhas criptografadas com bcrypt (salt rounds: 12)
- Tokens JWT seguros

Autorização

- Middleware protege rotas privadas
- Usuários só podem ver seus próprios registros (planejado)
- Admins têm acesso total

Boas Práticas

- ☒ HTTPS em produção (automático no Vercel)
- ☒ Variáveis de ambiente para secrets
- ☒ Validação de dados no backend
- ☒ SQL injection prevenido (Prisma ORM)
- ☒ XSS prevenido (React + NextJS)

Troubleshooting

Erro: “Prisma Client não inicializado”

```
npx prisma generate
```

Erro: “Não consegue conectar ao banco”

1. Verifique se PostgreSQL está rodando
2. Verifique credenciais no `.env`
3. Teste conexão: `psql $DATABASE_URL`

Erro: “Usuário não encontrado”

```
npm run prisma:seed
```

Página em branco após login

1. Limpe cookies do navegador
2. Verifique console do navegador (F12)
3. Verifique logs do servidor

Exportação não funciona

1. Verifique se há registros no banco
2. Abra DevTools (F12) e veja erros
3. Tente com navegador diferente



Relatórios e Análises

Estatísticas Disponíveis

- Total de registros
- Total de distância percorrida
- Média de distância por viagem
- Uso por veículo
- Uso por condutor

Nota: Dashboard de estatísticas em desenvolvimento



Casos de Uso

Empresa de Transporte

1. Registre todas as viagens dos motoristas
2. Exporte relatórios mensais
3. Monitore quilometragem para manutenção

Frota de Veículos

1. Controle uso de múltiplos veículos
2. Rastreie quem usou cada veículo
3. Gere relatórios para contabilidade

Uso Pessoal

1. Controle quilometragem do carro pessoal
2. Registre viagens a trabalho
3. Exporte para declaração de impostos

Atualizações

Verificar Versão

```
cat package.json | grep version
```

Atualizar Dependências

```
npm update
```

Migrations

```
# Criar nova migration
npx prisma migrate dev --name nome_da_migration

# Aplicar migrations em produção
npx prisma migrate deploy
```



Dicas e Truques

Atalhos de Teclado

- **Ctrl + K** : Busca rápida (planejado)
- **Ctrl + N** : Novo registro (planejado)
- **Escape** : Fechar modais

Performance

- Use filtros para limitar resultados
- Exporte apenas períodos necessários
- Limpe registros antigos periodicamente

Backup

```
# Backup do banco
pg_dump $DATABASE_URL > backup.sql

# Restaurar backup
psql $DATABASE_URL < backup.sql
```



Suporte

Documentação Adicional

- [README.md](#) (README.md) - Visão geral do projeto
- [DEPLOYMENT.md](#) (DEPLOYMENT.md) - Guia de deploy
- [Prisma Docs](https://www.prisma.io/docs) (https://www.prisma.io/docs)
- [NextJS Docs](https://nextjs.org/docs) (https://nextjs.org/docs)

Problemas Conhecidos

- [] Dashboard de estatísticas em desenvolvimento

- [] Filtros avançados em desenvolvimento
- [] Multi-tenancy em planejamento

Reportar Bugs

Abra uma issue no GitHub com:

1. Descrição do problema
2. Passos para reproduzir
3. Screenshots (se aplicável)
4. Logs de erro

Desenvolvido com ❤️ usando NextJS 14 + PostgreSQL + Prisma