



# Wayne Industries

## Sistema de Gerenciamento e Segurança

Documentação Técnica Completa

*Projeto Full Stack | Next.js 14 | TypeScript | Supabase*

🔗 Links do Projeto .....	4
📋 Sobre o Projeto .....	4
🎯 Objetivos do Projeto .....	4
🚀 Tecnologias Utilizadas .....	4
Frontend.....	4
Backend .....	4
🌟 Funcionalidades Principais.....	5
🔒 Autenticação e Segurança.....	5
📦 Gerenciamento de Recursos .....	5
📊 Dashboard Analítico .....	5
🛡️ Controle de Segurança.....	5
🗄️ Estrutura do Banco de Dados .....	6
1. user_profiles .....	6
2. resources .....	6
3. restricted_areas .....	6
4. access_logs .....	6
5. activities .....	6
🔒 Segurança Implementada .....	7
1. Row Level Security (RLS).....	7
2. Autenticação JWT .....	7
3. Middleware de Proteção.....	7
4. Validação de Dados .....	7
🔑 Credenciais de Teste .....	7
Administrador (Acesso Total).....	7
Gerente (Acesso Intermediário) .....	7
Funcionário (Acesso Básico).....	7
🛠️ Instalação e Configuração.....	8
Pré-requisitos .....	8
Passo 1: Clone o Repositório .....	8
Passo 2: Instale as Dependências.....	8
Passo 3: Configure o Supabase .....	8
Passo 4: Configure Variáveis de Ambiente.....	8
Passo 5: Execute o Projeto .....	8
📚 Aprendizados e Conceitos Aplicados .....	8
Frontend.....	8
Backend .....	8
Boas Práticas .....	9
🌟 Conclusão.....	10

Principais Conquistas ..... 10

 Informações do Projeto ..... 10

## Links do Projeto

Repositório GitHub: <https://github.com/Rhuan12/wayne-industries>

Deploy (Vercel): <https://wayne-industries-sigma.vercel.app/dashboard>

## Sobre o Projeto

O Wayne Industries System é uma aplicação web completa para gerenciamento de recursos e controle de segurança empresarial. Desenvolvido como projeto acadêmico, o sistema demonstra a implementação de um stack moderno de desenvolvimento web com foco em segurança, usabilidade e escalabilidade.

## Objetivos do Projeto

- Implementar autenticação segura com múltiplos níveis de acesso
- Desenvolver CRUD completo de recursos
- Criar dashboard com visualizações interativas
- Implementar sistema de controle de segurança e logs
- Aplicar boas práticas de desenvolvimento Full Stack

## Tecnologias Utilizadas

### Frontend

- **Next.js 14** - Framework React com App Router e Server Components
- **TypeScript** - Tipagem estática para maior segurança
- **Tailwind CSS** - Estilização utilitária e responsiva
- **Recharts** - Gráficos interativos e visualizações
- **React Hook Form** - Gerenciamento de formulários
- **Zod** - Validação de schemas
- **Sonner** - Notificações toast elegantes
- **Lucide React** - Ícones modernos

### Backend

#### Supabase - Backend as a Service (BaaS)

- PostgreSQL (Banco de dados)
- Authentication (JWT)
- Row Level Security (RLS)
- Real-time subscriptions

## Funcionalidades Principais

### Autenticação e Segurança

- Login seguro com email e senha
- 3 níveis de acesso (Admin, Manager, Employee)
- Proteção de rotas com middleware
- Row Level Security (RLS) no banco de dados
- Logout seguro com limpeza de sessão

### Gerenciamento de Recursos

- **CRUD Completo:** Create, Read, Update, Delete
- **Tipos de recursos:** Equipamentos, Veículos, Dispositivos de Segurança
- **Status:** Disponível, Em Uso, Manutenção, Aposentado
- **Filtros avançados:** Por tipo, status e busca textual
- **Validação de formulários:** Feedback em tempo real

### Dashboard Analítico

- Cards de estatísticas em tempo real
- Gráfico de barras (distribuição de recursos por tipo)
- Gráfico de linha (monitoramento de acessos)
- Lista de atividades recentes
- Indicadores visuais (cores e badges)

### Controle de Segurança

- 5 áreas restritas pré-cadastradas
- Logs de acesso (entrada, saída, negados)
- Controle de permissões por nível de acesso
- Estatísticas de segurança
- Histórico completo de atividades

# Estrutura do Banco de Dados

O sistema utiliza PostgreSQL com 5 tabelas principais:

## 1. user\_profiles

*Armazena perfis e permissões dos usuários*

- id (UUID) - Referência ao usuário do Supabase Auth
- email (TEXT)
- full\_name (TEXT)
- role (TEXT) - employee, manager, admin
- department (TEXT)

## 2. resources

*Recursos gerenciados pelo sistema*

- id (UUID)
- type (TEXT) - equipment, vehicle, security\_device
- name (TEXT)
- status (TEXT) - available, in\_use, maintenance, retired
- location (TEXT)
- created\_by (UUID) - Referência ao usuário

## 3. restricted\_areas

*Áreas com acesso controlado*

- id (UUID)
- name (TEXT)
- required\_access\_level (TEXT) - employee, manager, admin
- is\_active (BOOLEAN)

## 4. access\_logs

*Registros de acesso às áreas*

- user\_id (UUID)
- area\_id (UUID)
- action (TEXT) - entry, exit, denied
- timestamp (TIMESTAMP)

## 5. activities

*Histórico de ações no sistema*

- user\_id (UUID)
- resource\_id (UUID) - Opcional
- action\_type (TEXT)
- description (TEXT)
- timestamp (TIMESTAMP)

# Segurança Implementada

## 1. Row Level Security (RLS)

Todas as tabelas possuem políticas RLS ativas que garantem:

- Usuários só acessam dados permitidos pelo seu nível
- Logs são protegidos contra modificação
- Recursos só podem ser editados pelo criador (ou admin)

## 2. Autenticação JWT

- Tokens seguros fornecidos pelo Supabase Auth
- Refresh automático de sessão
- Logout com limpeza completa

## 3. Middleware de Proteção

- Rotas protegidas por autenticação
- Redirecionamento automático
- Verificação de sessão em cada requisição

## 4. Validação de Dados

- Validação client-side com Zod
- Validação server-side no Supabase
- Sanitização de inputs

# Credenciais de Teste

### Administrador (Acesso Total)

- **Email:** admin@wayne.com
- **Senha:** admin123
- **Permissões:** Todas as funcionalidades

### Gerente (Acesso Intermediário)

- **Email:** manager@wayne.com
- **Senha:** manager123
- **Permissões:** Visualizar e gerenciar recursos

### Funcionário (Acesso Básico)

- **Email:** employee@wayne.com
- **Senha:** employee123
- **Permissões:** Visualizar recursos e áreas permitidas

# Instalação e Configuração

## Pré-requisitos

- Node.js 18+ instalado
- npm ou yarn
- Conta no Supabase (gratuita)
- Git

## Passo 1: Clone o Repositório

```
git clone https://github.com/Rhuan12/wayne-industries.git  
cd wayne-industries
```

## Passo 2: Instale as Dependências

```
npm install
```

## Passo 3: Configure o Supabase

1. Acesse [supabase.com](https://supabase.com)
2. Crie uma conta (se não tiver)
3. Clique em 'New Project'
4. Preencha o nome do projeto: wayne-industries

## Passo 4: Configure Variáveis de Ambiente

Crie um arquivo .env.local na raiz do projeto:

```
NEXT_PUBLIC_SUPABASE_URL=https://seu-projeto.supabase.co  
NEXT_PUBLIC_SUPABASE_ANON_KEY=sua-chave-anon  
NEXT_PUBLIC_APP_URL=http://localhost:3000
```

## Passo 5: Execute o Projeto

```
npm run dev
```

Acesse: <http://localhost:3000>

## Aprendizados e Conceitos Aplicados

### Frontend

- Next.js App Router e Server Components
- TypeScript para tipagem forte
- React Hooks (useState, useRouter, usePathname)
- Form handling com React Hook Form
- Schema validation com Zod
- Componentização e reusabilidade
- Tailwind CSS e design system

### Backend

- Supabase como BaaS

- PostgreSQL e modelagem de dados
- Row Level Security (RLS)
- Autenticação e autorização
- Queries e filtros complexos

## Boas Práticas

- Separação de concerns (components, lib, types)
- Client vs Server Components
- Middleware para proteção de rotas
- Tratamento de erros
- Loading states e feedback visual
- Código limpo e documentado

## ❖ Conclusão

O Wayne Industries System demonstra com sucesso a implementação de um sistema full stack moderno, aplicando tecnologias de ponta e melhores práticas de desenvolvimento web. O projeto atende a todos os requisitos propostos e oferece uma base sólida para expansões futuras.

## Principais Conquistas

- Sistema 100% funcional e deployado
- Arquitetura escalável e segura
- Interface moderna e responsiva
- Código limpo e bem documentado
- Implementação completa de controle de acesso

## 💻 Informações do Projeto

**Autor:** Rhuan Mateus

**GitHub:** [@Rhuan12](#)

**Repositório:** [wayne-industries](#)

**Deploy:** <https://wayne-industries-sigma.vercel.app/dashboard>

---

*Desenvolvido com 🚀 por Rhuan Mateus*

Wayne Industries System © 2026