

Sistema de Gestão de Pessoas - HR Management

Um sistema completo de gestão de recursos humanos desenvolvido com tecnologias modernas, oferecendo uma solução integrada para administração de funcionários, avaliações de desempenho, benefícios, treinamentos e muito mais.

Índice

- [Visão Geral](#)
- [Funcionalidades](#)
- [Tecnologias Utilizadas](#)
- [Arquitetura](#)
- [Instalação](#)
- [Configuração](#)
- [Uso](#)
- [API Documentation](#)
- [Banco de Dados](#)
- [Deploy](#)
- [Contribuição](#)
- [Licença](#)

Visão Geral

O Sistema de Gestão de Pessoas é uma plataforma web moderna e intuitiva que centraliza todas as operações de recursos humanos em uma única aplicação. Desenvolvido com foco na experiência do usuário e na eficiência operacional, o sistema oferece interfaces responsivas e funcionalidades avançadas para gestão completa de pessoas.

Principais Características

- **Interface Moderna:** Design responsivo inspirado nas melhores práticas de UX/UI
- **Chat Inteligente:** Assistente virtual integrado para suporte aos funcionários
- **Dashboard Analítico:** Visualizações e métricas em tempo real
- **Segurança Avançada:** Autenticação JWT e controle de acesso baseado em roles
- **Escalabilidade:** Arquitetura modular preparada para crescimento

- **Integração:** APIs RESTful para integração com sistemas externos

Funcionalidades

Gestão de Funcionários

- Cadastro completo de colaboradores
- Perfis detalhados com informações pessoais e profissionais
- Controle de status (ativo/inativo)
- Histórico de alterações

Estrutura Organizacional

- Gestão de departamentos
- Definição de cargos e hierarquias
- Atribuição de gestores
- Organograma dinâmico

Avaliações de Desempenho

- Ciclos de avaliação personalizáveis
- Múltiplos critérios de avaliação
- Feedback 360 graus
- Relatórios de performance

Treinamentos e Desenvolvimento

- Catálogo de cursos e treinamentos
- Inscrições e acompanhamento
- Certificações
- Planos de desenvolvimento individual

Gestão de Férias

- Solicitação online de férias
- Aprovação por gestores
- Calendário de férias
- Controle de saldo

Benefícios

- Catálogo de benefícios disponíveis

- Adesão e cancelamento
- Controle de custos
- Relatórios de utilização

Chat e Comunicação

- Chat em tempo real entre funcionários
- Assistente virtual (bot) para dúvidas de RH
- Notificações push
- Histórico de conversas

Relatórios e Analytics

- Dashboard executivo
- Relatórios personalizáveis
- Métricas de RH
- Exportação de dados

Tecnologias Utilizadas

Backend

- **Node.js** - Runtime JavaScript
- **NestJS** - Framework backend progressivo
- **TypeScript** - Linguagem tipada
- **Prisma** - ORM moderno
- **PostgreSQL** - Banco de dados relacional
- **JWT** - Autenticação e autorização
- **Socket.io** - Comunicação em tempo real
- **Swagger** - Documentação de API

Frontend

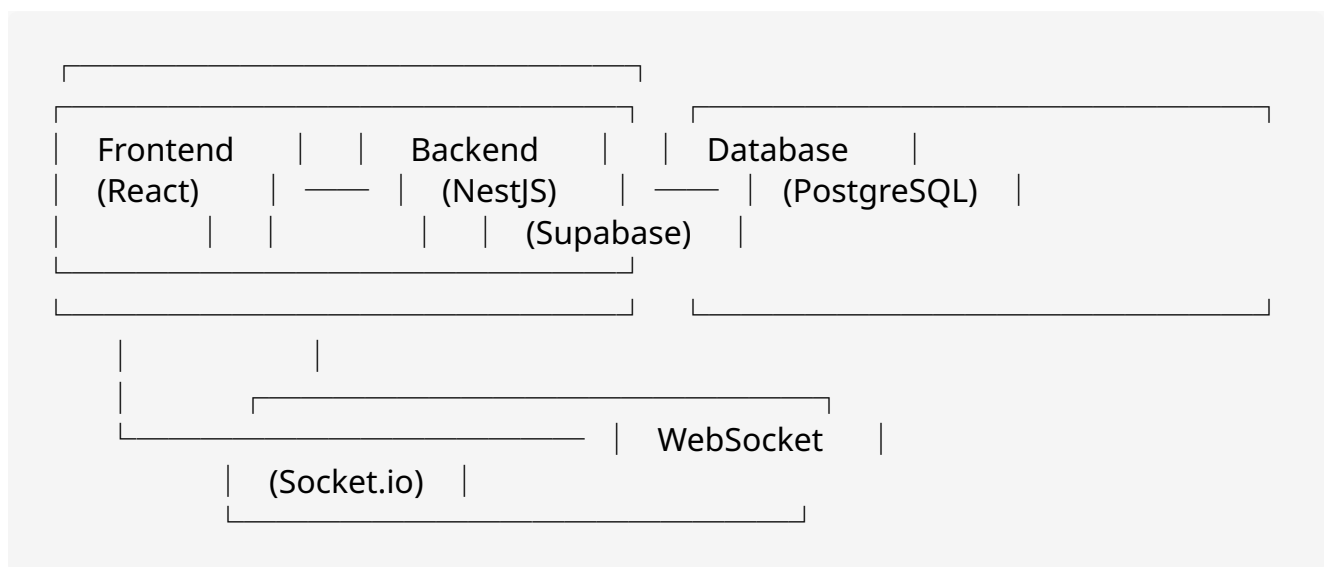
- **React** - Biblioteca para interfaces
- **TypeScript** - Linguagem tipada
- **Vite** - Build tool moderna
- **Tailwind CSS** - Framework CSS utilitário
- **shadcn/ui** - Componentes UI modernos
- **Recharts** - Gráficos e visualizações
- **React Router** - Roteamento
- **Axios** - Cliente HTTP

Infraestrutura

- **Supabase** - Backend as a Service
- **Docker** - Containerização
- **GitHub Actions** - CI/CD
- **Vercel/Netlify** - Deploy frontend
- **Railway/Heroku** - Deploy backend

Arquitetura

O sistema segue uma arquitetura moderna de microserviços com separação clara entre frontend e backend:



Camadas da Aplicação

1. **Apresentação (Frontend)**
2. Componentes React reutilizáveis
3. Gerenciamento de estado com Context API
4. Roteamento client-side
5. Interface responsiva
6. **Lógica de Negócio (Backend)**
7. Controllers para endpoints REST
8. Services para lógica de negócio
9. Guards para autenticação/autorização
10. Middlewares para validação
11. **Persistência (Database)**

- 12. Modelos Prisma
- 13. Migrações versionadas
- 14. Índices otimizados
- 15. Backup automatizado

Instalação

Pré-requisitos

- Node.js 18+
- npm ou yarn
- PostgreSQL 14+
- Git

Clonando o Repositório

```
git clone https://github.com/seu-usuario/hr-management-system.git  
cd hr-management-system
```

Configuração do Backend

```
cd backend  
npm install  
cp .env.example .env  
# Configure as variáveis de ambiente no .env  
npm run prisma:generate  
npm run prisma:migrate  
npm run dev
```

Configuração do Frontend

```
cd hr-management-frontend  
npm install  
cp .env.example .env  
# Configure a URL da API no .env  
npm run dev
```

Configuração

Variáveis de Ambiente

Backend (.env)

```
DATABASE_URL="postgresql://user:password@localhost:5432/hr_management"
JWT_SECRET="seu-jwt-secret-super-seguro"
JWT_EXPIRES_IN="7d"
PORT=3000
NODE_ENV="development"
CORS_ORIGIN="http://localhost:5173"
```

Frontend (.env)

```
VITE_API_URL=http://localhost:3000/api/v1
VITE_WS_URL=http://localhost:3000
```

Configuração do Banco de Dados

1. **Supabase Setup**
2. Crie um projeto no [Supabase](#)
3. Execute os scripts SQL fornecidos na pasta `database/`
4. Configure a string de conexão

5. Migrações Locais

```
bash
```

```
npm run prisma:migrate
```

```
npm run prisma:seed
```

Uso

Acesso ao Sistema

1. **Desenvolvimento**
2. Frontend: `http://localhost:5173`
3. Backend: `http://localhost:3000`
4. API Docs: `http://localhost:3000/api`
5. **Credenciais Padrão**

6. Email: admin@hrmanagement.com

7. Senha: admin123

Fluxo de Trabalho

1. **Login:** Acesse com suas credenciais
2. **Dashboard:** Visualize métricas e informações gerais
3. **Navegação:** Use o menu lateral para acessar diferentes módulos
4. **Chat:** Clique no ícone de chat para interagir com o assistente virtual
5. **Perfil:** Gerencie suas informações pessoais

Documentação Adicional

- [Arquitetura Detalhada](#)
- [API Reference](#)
- [Guia de Instalação](#)
- [Configuração do Banco](#)
- [Deploy em Produção](#)
- [Guia de Contribuição](#)

Contribuição

Contribuições são bem-vindas! Por favor, leia o [guia de contribuição](#) antes de submeter pull requests.

Processo de Desenvolvimento

1. Fork o projeto
2. Crie uma branch para sua feature (`git checkout -b feature/AmazingFeature`)
3. Commit suas mudanças (`git commit -m 'Add some AmazingFeature'`)
4. Push para a branch (`git push origin feature/AmazingFeature`)
5. Abra um Pull Request

Licença

Este projeto está licenciado sob a Licença MIT - veja o arquivo [LICENSE](#) para detalhes.

Suporte

Para suporte e dúvidas:

- Email: suporte@hrmanagement.com
 - Issues: [GitHub Issues](#)
 - Documentação: [Wiki](#)
-

Desenvolvido com pela equipe de desenvolvimento