

P2Paga - Sistema de Gerenciamento de Cobranças

📄 Sobre o Sistema

O P2Paga é um sistema completo de gerenciamento de cobranças desenvolvido para empresas que precisam gerenciar seus clientes e cobranças de forma eficiente. O sistema oferece uma interface moderna e intuitiva, com recursos automatizados para controle de vencimentos e comunicação com clientes.

🚀 Funcionalidades Principais

👤 Gestão de Clientes

- Cadastro completo de clientes
- Informações de contato (email, telefone, WhatsApp)
- Endereçamento
- Histórico de cobranças por cliente

💰 Gestão de Cobranças

- Criação e gerenciamento de cobranças
- Status automático (pendente, pago, atrasado)
- Verificação automática de vencimentos
- Histórico de pagamentos
- Relatórios e estatísticas

🔒 Sistema de Autenticação

- Login para empresas
- Login para administradores
- Proteção de rotas
- Gerenciamento de permissões

📊 Dashboard e Relatórios

- Visão geral das cobranças
- Estatísticas de pagamentos
- Relatórios personalizados
- Gráficos e indicadores

📱 Comunicação

- Integração com WhatsApp (simulada)
- Notificações automáticas
- Alertas de vencimento

🛠️ Tecnologias Utilizadas

Frontend

- React 18
- TypeScript
- Vite
- shadcn-ui
- Tailwind CSS
- React Query
- React Router DOM

Backend

- Node.js
- Express
- MySQL
- JWT para autenticação
- Node Schedule para tarefas agendadas

📦 Instalação

Pré-requisitos

- Node.js (versão LTS recomendada)
- MySQL
- npm ou yarn

Passos para Instalação

1. Clone o repositório:

```
`` `sh  
git clone https://github.com/seu-usuario/P2Paga.git  
cd P2Paga  
` ` `
```

2. Instale as dependências do frontend:

```
`` `sh  
npm install  
` ` `
```

3. Configure o banco de dados:

```
`` `sh  
# Crie um arquivo .env na raiz do projeto com as seguintes variáveis:
```

```
DB_HOST=localhost  
DB_USER=seu_usuario  
DB_PASSWORD=sua_senha  
DB_NAME=sistema_cobrancas  
JWT_SECRET=sua_chave_secreta  
` ` `
```

4. Instale e configure o backend:

```
```sh
cd src/server
npm install
```
```

5. Execute o script de instalação do banco de dados:

```
```sh
node install-db.js
```
```

🚀 Executando o Sistema

1. Inicie o backend:

```
```sh
cd src/server
npm start
```
```

2. Em outro terminal, inicie o frontend:

```
```sh
npm run dev
```
```

3. Acesse o sistema em `http://localhost:8080``

📌 Rotas da API

Todas as rotas da API são prefixadas com `/api`. Exemplo:
`http://localhost:5000/api/auth/login`

🔑 Autenticação

...

POST /api/auth/login

Body: { email: string, senha: string }

...

👤 Clientes

...

GET /api/clientes # Listar todos os clientes

GET /api/clientes/:id # Buscar cliente específico

POST /api/clientes # Criar novo cliente

PUT /api/clientes/:id # Atualizar cliente

DELETE /api/clientes/:id # Excluir cliente

POST /api/clientes/:id/notificar # Enviar notificação ao cliente

...

💰 Cobranças

...

GET /api/cobrancas # Listar todas as cobranças

GET /api/cobrancas/:id # Buscar cobrança específica

POST /api/cobrancas # Criar nova cobrança

PUT /api/cobrancas/:id # Atualizar cobrança

DELETE /api/cobrancas/:id # Excluir cobrança

PATCH /api/cobrancas/:id/status # Atualizar status da cobrança
GET /api/cobrancas/clientes/:clienteId/cobrancas # Listar cobranças por cliente
POST /api/cobrancas/clientes/:clienteId/cobrancas # Criar cobrança para cliente
POST /api/cobrancas/verificar-vencidas # Verificar cobranças vencidas
...

🏢 Empresas (Apenas Admin)

...

GET /api/empresas # Listar todas as empresas
GET /api/empresas/:id # Buscar empresa específica
POST /api/empresas # Criar nova empresa
PUT /api/empresas/:id # Atualizar empresa
DELETE /api/empresas/:id # Excluir empresa
GET /api/empresas/estatisticas # Obter estatísticas das empresas
...

📱 Notificações

...

POST /api/notificacoes/enviar # Enviar notificação

...

🔒 Autenticação e Autorização

- Todas as rotas (exceto login) requerem token JWT no header:

...

Authorization: Bearer <seu_token>

...

- Empresas só podem acessar seus próprios dados

- Admins têm acesso a todos os dados

📄 Exemplo de Uso

```
```javascript
// Exemplo de requisição para criar uma cobrança
fetch('http://localhost:5000/api/cobrancas', {
 method: 'POST',
 headers: {
 'Content-Type': 'application/json',
 'Authorization': 'Bearer seu_token_jwt'
 },
 body: JSON.stringify({
 clienteId: '123',
 descricao: 'Fatura Mensal',
 valor: 199.90,
 dataVencimento: '2024-03-30'
 })
});
```
```

🔄 Verificações Automáticas

O sistema realiza verificações automáticas de cobranças vencidas:

- A cada 5 minutos durante o uso do sistema
- Diariamente às 00:01
- Ao iniciar o sistema

🛡️ Segurança

- Autenticação via JWT
- Proteção de rotas
- Validação de dados
- Credenciais seguras via variáveis de ambiente
- CORS configurado para desenvolvimento

📄 Estrutura do Projeto

...

P2Paga/

```
├─ src/  
|   ├─ components/ # Componentes React  
|   ├─ contexts/   # Contextos da aplicação  
|   ├─ hooks/      # Hooks personalizados  
|   ├─ pages/      # Páginas da aplicação  
|   ├─ services/   # Serviços e APIs  
|   └─ server/     # Backend  
|       ├─ config/ # Configurações  
|       ├─ controllers/# Controladores  
|       └─ models/ # Modelos  
└─ routes/ # Rotas da API  
├─ public/    # Arquivos estáticos  
└─ ...
```

...

📄 Licença

Este projeto está sob a licença ISC. Veja o arquivo `LICENSE` para mais detalhes.