

NEXUS Process Solutions

Equipe: Thiago Henrique

Luís Carlos

Endi Rayanna

Felipe Cavalcanti

PROJETO DE GESTÃO DE PROCESSOS

SISTEMA PROCESSNET DOCUMENTAÇÃO BACKEND

Visão Geral do Sistema

O **ProcessNet** é um sistema de gestão de processos desenvolvido pela **Nexus Process Solutions**. Ele permite o cadastro e gerenciamento de processos de vistoria e análise, bem como a consulta e atualização do status desses processos por diferentes atores (Usuários, Vistoriadores, Analistas, Gestores).

Arquitetura do Sistema

O sistema é dividido em várias camadas, cada uma responsável por diferentes aspectos da aplicação:

1. **Modelos de Dados:** Representam a estrutura dos dados do sistema.
2. **Repositórios:** Interface com o banco de dados.
3. **Serviços:** Contêm a lógica de negócio.
4. **Controladores:** Exposição dos serviços através de endpoints REST.

Modelos de Dados

`PessoaModel`

- Campos: `idPessoa`, `nome`, `cpf`, `email`, `senha`, `telefone`
- Descrição: Classe base para `UsuarioModel` e `FuncionarioModel`.

`UsuarioModel`

- Herda de `PessoaModel`.
- Descrição: Representa os usuários do sistema.

`FuncionarioModel`

- Herda de `PessoaModel`.
- Campos adicionais: `cargo`
- Descrição: Representa os funcionários do sistema com diferentes cargos (Vistoriador, Analista, Gestor).

`ProcessoModel`

- Campos: `id`, `numeroProtocolo`, `dataCriacao`, `tipoProcesso`, `cnpj`, `endereco`, `responsavel`, `status`, `parecer`
- Descrição: Representa os processos de vistoria e análise.

`LoginModel`

- Campos: `cpf`, `senha`
- Descrição: Usado para autenticação de usuários e funcionários.

Repositórios

`PessoaRepository`

- Métodos principais: `findByCpf`
- Descrição: Interface para operações CRUD com a entidade `PessoaModel`.

`UsuarioRepository`

- Herda de `PessoaRepository`.
- Métodos principais: `findByCpf`
- Descrição: Interface para operações CRUD com a entidade `UsuarioModel`.

`FuncionarioRepository`

- Herda de `PessoaRepository`.
- Métodos principais: `findByCargo`, `findByCpf`
- Descrição: Interface para operações CRUD com a entidade `FuncionarioModel`.

`ProcessoRepository`

- Métodos principais: `findByResponsavel_IdPessoa`, `findByTipoProcesso`, `findByStatus`
- Descrição: Interface para operações CRUD com a entidade `ProcessoModel`.

Serviços

`PessoaService`

- Métodos principais: `create`, `findAll`, `findById`, `update`
- Descrição: Lógica de negócio para operações com a entidade `PessoaModel`.

`UsuarioService`

- Herda de `PessoaService`.
- Métodos adicionais: `delete`.
- Descrição: Lógica de negócio para operações com a entidade `UsuarioModel`.

`FuncionarioService`

- Herda de `PessoaService`.
- Métodos adicionais: `delete`, `findByCargo`.
- Descrição: Lógica de negócio para operações com a entidade `FuncionarioModel`.

`ProcessoService`

- Métodos principais: `create`, `findByResponsavel`, `findByTipoProcesso`, `findByStatus`, `updateStatus`, `updateParecer`, `selectProcesso`, `findAll`, `findById`.
- Descrição: Lógica de negócio para operações com a entidade `ProcessoModel`.

`LoginService`

- Métodos principais: `authenticateUsuario`, `authenticateFuncionario`.
- Descrição: Lógica de autenticação de usuários e funcionários.

Controladores

`UsuarioController`

- Endpoints principais: `GET /usuarios`, `GET /usuarios/{id}`, `POST /usuarios`, `POST /usuarios/authenticate`, `PUT /usuarios/{id}`, `DELETE /usuarios/{id}`.
- Descrição: Controlador para operações com usuários.

`FuncionarioController`

- Endpoints principais: `GET /funcionarios`, `GET /funcionarios/{id}`, `POST /funcionarios`, `POST /funcionarios/authenticate`, `PUT /funcionarios/{id}`, `DELETE /funcionarios/{id}`.
- Descrição: Controlador para operações com funcionários.

`ProcessoController`

- Endpoints principais: `GET /processos`, `GET /processos/{id}`, `POST /processos`, `PUT /processos/{id}/status`, `PUT /processos/{id}/parecer`, `GET /processos/responsavel/{idPessoa}`, `GET /processos/tipo/{tipo}`, `GET /processos/status/{status}`.
- Descrição: Controlador para operações com processos.

Casos de Uso

1. Cadastro de Usuários e Funcionários:

- Descrição: Usuários e funcionários podem cadastrar-se no sistema fornecendo detalhes como nome completo, CPF, e-mail e telefone. Funcionários indicam sua função.

- Fluxo principal: O usuário acessa a página de cadastro, preenche os campos necessários, submete o formulário, o sistema valida os dados e registra o novo usuário ou funcionário.

2. Login no Sistema:

- Descrição: Usuários e funcionários acessam o sistema usando CPF e senha.

- Fluxo principal: O usuário insere CPF e senha, o sistema verifica as credenciais e concede o acesso.

3. Cadastro de Solicitações de Processo:

- Descrição: Usuários podem iniciar novos processos de vistoria ou análise.

- Fluxo principal: O usuário acessa a seção de novo processo, preenche as informações necessárias e submete a solicitação, o sistema gera um número de protocolo e registra o novo processo.

4. Consulta de Processos:

- Descrição: Diferentes atores podem consultar o status de processos usando número do processo ou CNPJ.

- Fluxo principal: O ator acessa a seção de consulta, insere o número do processo ou CNPJ, o sistema busca e exibe o status do processo.

5. Consulta e Alteração de Status de Processos por Vistoriador:

- Descrição: Vistoriador consulta e altera status de processos de vistoria.

- Fluxo principal: Vistoriador acessa a lista de processos de vistoria, seleciona um processo, o sistema exibe uma interface para escolha do novo status, Vistoriador escolhe o status e confirma a alteração.

6. Consulta e Alteração de Status de Processos por Analista:

- Descrição: Analista consulta e altera status de processos de análise.

- Fluxo principal: Analista acessa a lista de processos de análise, seleciona um processo, o sistema exibe uma interface para escolha do novo status, Analista escolhe o status e confirma a alteração.

7. Consulta de Processos por Gestor:

- Descrição: Gestor visualiza o status de todos os processos.

- Fluxo principal: Gestor acessa o painel de controle, visualiza as colunas de processos com seus respectivos status.

Diagrama de Classes

Para uma visão detalhada do relacionamento entre as classes, consulte o diagrama de classes no arquivo fornecido.

Endpoints da API

Usuários

- `GET /usuarios``: Retorna todos os usuários.
- `GET /usuarios/{id}``: Retorna um usuário pelo ID.
- `POST /usuarios``: Cria um novo usuário.
- `POST /usuarios/authenticate``: Autentica um usuário.
- `PUT /usuarios/{id}``: Atualiza um usuário pelo ID.
- `DELETE /usuarios/{id}``: Deleta um usuário pelo ID.

Funcionários

- `GET /funcionarios``: Retorna todos os funcionários.
- `GET /funcionarios/{id}``: Retorna um funcionário pelo ID.
- `POST /funcionarios``: Cria um novo funcionário.
- `POST /funcionarios/authenticate``: Autentica um funcionário.
- `PUT /funcionarios/{id}``: Atualiza um funcionário pelo ID.
- `DELETE /funcionarios/{id}``: Deleta um funcionário pelo ID.

Processos

- `GET /processos``: Retorna todos os processos.
- `GET /processos/{id}``: Retorna um processo pelo ID.
- `POST /processos``: Cria um novo processo.
- `PUT /processos/{id}/status``: Atualiza o status de um processo.
- `PUT /processos/{id}/parecer``: Atualiza o parecer de um processo.
- `GET /processos/responsavel/{idPessoa}``: Retorna processos pelo ID do responsável.
- `GET /processos/tipo/{tipo}``: Retorna processos pelo tipo.
- `GET /processos/status/{status}``: Retorna processos pelo status.