Documentação do Sistema de Gestão de Patrimônio da FACOM-UFMS

Visão Geral do Projeto

O Sistema de Gestão Patrimonial FACOM/UFMS é uma aplicação web completa desenvolvida para o registro, controle e gerenciamento de bens patrimoniais da Faculdade de Computação da Universidade Federal de Mato Grosso do Sul (FACOM/UFMS).

Propósito

- Centralizar o controle de patrimônio da FACOM
- Facilitar o registro e determinação da localização de itens patrimoniais
- Gerenciar empréstimos e movimentações de equipamentos
- Gerar relatórios e termos de corresponsabilidade
- Manter histórico completo de movimentações

Principais Funcionalidades

✅ Gestão de Itens: Cadastro, edição e consulta de bens patrimoniais

✅ Controle de Movimentações: Alocação, transferência e empréstimos

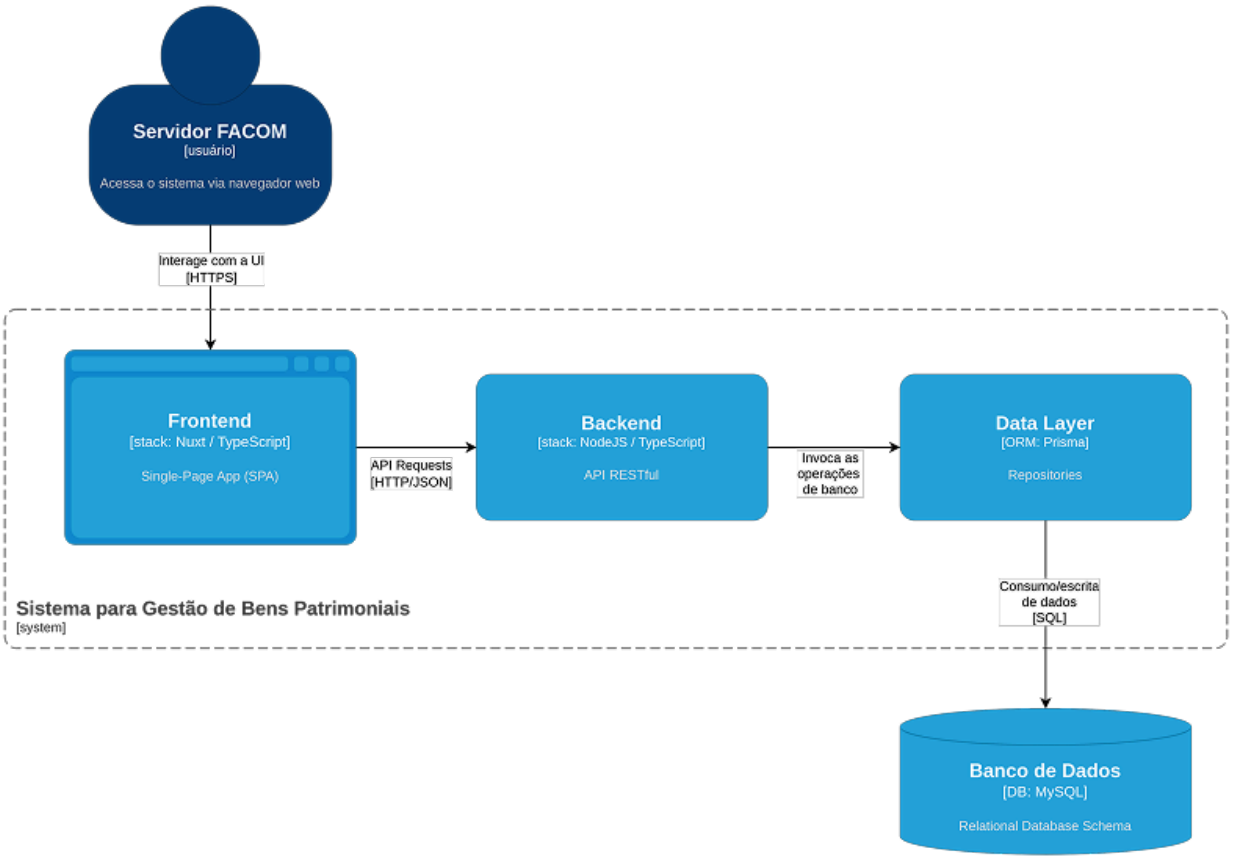
✅ Gestão de Locais: Controle de salas e responsáveis

✅ Relatórios: Exportação em Excel e geração de termos

✅ Dashboard: Visão geral com estatísticas e ações rápidas

✅ Histórico Completo: Timeline de todas as movimentações

Arquitetura do Sistema



Instalação e Configuração

- Pré-requisitos
  - Node.js (versão 18 ou superior)
  - MySQL

Dependências Principais

```
{
  "nuxt": "^4.1.2",
  "vue": "^3.5.22",
  "@prisma/client": "^6.17.1",
  "@nuxt/ui": "^4.1.0",
  "typescript": "^5.9.3",
```

```
  "@pinia/nuxt": "^0.11.3",
  "prisma": "^6.17.1"
}
```

Passos de Instalação

1. Clone o repositório
2. Instale as dependências

```
npm install
# ou
pnpm install
# ou
yarn install
```

3. Configure o banco de dados

```
# Configure as variáveis de ambiente no arquivo .env
DATABASE_URL="mysql://usuario:senha@localhost:3306/patrimonio_facom_db"
```

4. Execute as migrações do banco

```
npx prisma migrate dev --name init
```

5. Importação de Dados

```
npm run import-csv scripts/caminho/arquivo.csv    # Importa itens via CSV
```

5. Execute os seeds (opcional)

```
npm run seed:all
```

6. Inicie o servidor de desenvolvimento

```
npm run dev
```

O sistema estará disponível em http://localhost:3000

Organização Geral do Repositório

```
projeto/
├── app/                    # Frontend (Nuxt 3 + Vue 3)
│   ├── components/         # Componentes reutilizáveis
│   ├── composables/        # Composables Vue
│   ├── interfaces/         # Definições de tipos TypeScript
│   ├── pages/              # Páginas da aplicação
│   ├── services/           # Serviços para comunicação com API
│   ├── @stores/            # Stores Pinia para gerenciamento de estado
│   └── templates/          # Templates para geração de documentos
├── server/                 # Backend (Nuxt Server API)
│   ├── api/                # Endpoints da API
│   ├── interfaces/         # Interfaces do backend
│   ├── repositories/       # Camada de acesso a dados
│   ├── services/           # Lógica de negócio
│   └── utils/              # Utilitários do servidor
├── prisma/                 # Configuração do Prisma ORM
│   ├── schema.prisma       # Schema do banco de dados
│   └── seeds/              # Scripts de população inicial
└── scripts/                # Scripts utilitários
    └── init-items-data/    # Scripts de importação de dados
```

Módulo Frontend (app/)

Páginas Principais

`app/pages/index.vue`: Dashboard principal com estatísticas e ações rápidas

`app/pages/itens/index.vue`: Listagem e gerenciamento de itens patrimoniais
`app/pages/itens/[id].vue`: Página de detalhes e histórico de um item específico
`app/pages/salas/index.vue`: Gerenciamento de salas e locais

Componentes Importantes

`app/pages/itens/components/ItemsTable.vue`: Tabela principal de itens com ações
`app/pages/itens/components/CreateItemModal.vue`: Modal para cadastro de novos itens
`app/pages/itens/components/ItemDetails.vue`: Exibição detalhada de informações do item
`app/pages/itens/components/ItemHistory.vue`: Timeline do histórico de movimentações

Stores (Gerenciamento de Estados Globais)

`app/@stores/useItemStore.ts`: Estado global dos itens patrimoniais
`app/@stores/useOfficeStore.ts`: Estado global das salas e locais
`app/@stores/usePlacementStore.ts`: Estado global dos locais
`app/@stores/useAssigneeStore.ts`: Estado global dos responsáveis

Serviços

`app/services/item.service.ts`: Comunicação com API de itens
`app/services/office.service.ts`: Comunicação com API de salas
`app/services/export.service.ts`: Serviços de exportação de relatórios

Templates de Documentos

`app/templates/StudentLoanTerm.vue`: Termo de empréstimo para estudantes
`app/templates/EmployeeLoanTerm.vue`: Termo de empréstimo para funcionários
`app/templates/OfficesTerm.vue`: Termo de corresponsabilidade de salas

Módulo Backend (server/**)

API Endpoints

`server/api/item/`: CRUD de itens patrimoniais

- GET /api/item - Listagem de itens
- POST /api/item - Criação de item
- GET /api/item/[id] - Busca item por ID ou NFP
- GET /api/item/export - Exportação de relatórios

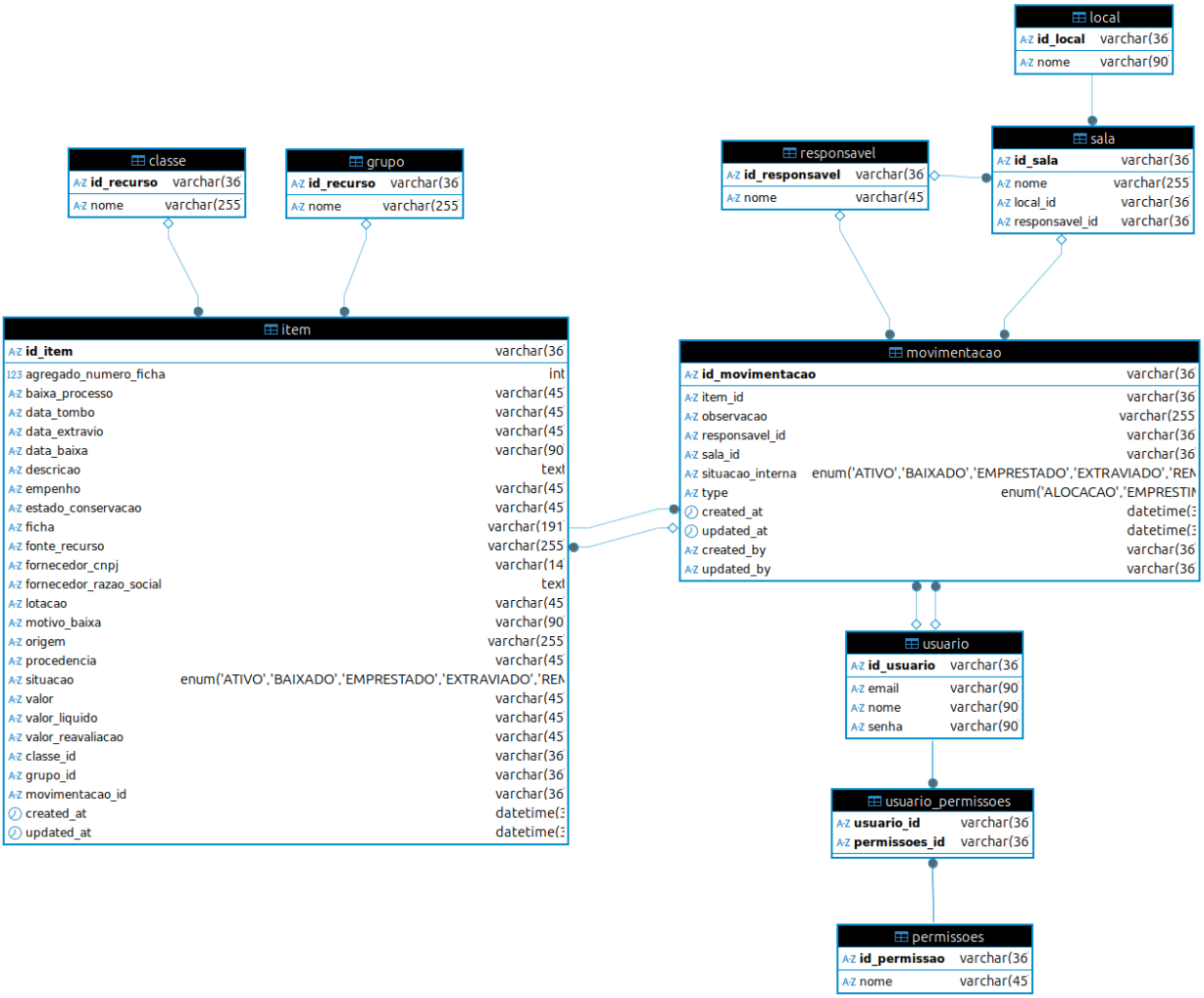
`server/api/office/`: CRUD de salas
`server/api/placement/`: CRUD de locais
`server/api/assignee/`: CRUD de responsáveis
`server/api/movement/`: CRUD de movimentações

Camadas da Arquitetura

- Services (Lógica de Negócio)
  - server/services/item.service.ts: Regras de negócio para itens
  - server/services/movement.service.ts: Regras de negócio para movimentações
  - server/services/office.service.ts: Regras de negócio para salas
  - server/services/export.service.ts: Lógica de exportação de dados
- Repositories (Acesso a Dados)
  - server/repositories/item.repository.ts: Operações de banco para itens
  - server/repositories/movement.repository.ts: Operações de banco para movimentações
  - server/repositories/office.repository.ts: Operações de banco para salas
- Interfaces
  - server/interfaces/item.interfaces.ts: Tipos e interfaces para itens
  - server/interfaces/movement.interfaces.ts: Tipos e interfaces para movimentações
  - server/interfaces/shared.interfaces.ts: Interfaces com

Banco de Dados

Schema Principal: `prisma/schema.prisma`


local
id_local varchar(36
nome varchar(90
sala
id_sala varchar(36
nome varchar(255
local_id varchar(36
responsavel_id varchar(36
responsavel
id_responsavel varchar(36
nome varchar(45
classe
id_recurso varchar(36
nome varchar(255
grupo
id_recurso varchar(36
nome varchar(255
item
id_item varchar(36
agregado_numero_ficha int
baixa_processo varchar(45
data_tombo varchar(45
data_extravio varchar(45
data_baixa varchar(90
descricao text
empenho varchar(45
estado_conservacao varchar(45
ficha varchar(191
fonte_recurso varchar(255
fornecedor_cnpj varchar(14
fornecedor_razao_social text
lotacao varchar(45
motivo_baixa varchar(90
origem varchar(255
procedencia varchar(45
situacao enum('ATIVO','BAIXADO','EMPRESTADO','EXTRAVIADO','REM
valor varchar(45
valor_liquido varchar(45
valor_reavaliacao varchar(45
classe_id varchar(36
grupo_id varchar(36
movimentacao_id varchar(36
created_at datetime(
updated_at datetime(
movimentacao
id_movimentacao varchar(36
item_id varchar(36
observacao varchar(255
responsavel_id varchar(36
sala_id varchar(36
situacao_interna enum('ATIVO','BAIXADO','EMPRESTADO','EXTRAVIADO','REM
type enum('ALOCACAO','EMPRESTI
created_at datetime(
updated_at datetime(
created_by varchar(36
updated_by varchar(36
usuario
id_usuario varchar(36
email varchar(90
nome varchar(90
senha varchar(90
usuario_permissoes
usuario_id varchar(36
permissoes_id varchar(36
permissoes
id_permissao varchar(36
nome varchar(45