

Documentação Técnica e Pedagógica

Sistema Web – Cadastro de Patrimônio

Avaliação Prática – Aula Teste

Cargo: Instrutor de Formação Profissional III – Tecnologia da Informação

Instituição: SENAI

Este documento foi elaborado com caráter técnico e didático, visando explicar não apenas o funcionamento do sistema, mas também os conceitos de desenvolvimento web envolvidos.

1. Contextualização do Projeto

O Sistema Web de Cadastro de Patrimônio foi desenvolvido para demonstrar, de forma prática, como uma aplicação web moderna pode ser estruturada utilizando separação entre back-end e front-end. O contexto do sistema simula uma situação real de empresas ou instituições que precisam controlar seus bens patrimoniais de maneira organizada, segura e eficiente.

2. Objetivo Educacional

Do ponto de vista educacional, o projeto tem como objetivo demonstrar competências relacionadas a: lógica de programação, organização de código, consumo de API REST, modelagem de banco de dados e desenvolvimento de interfaces web responsivas.

3. Tecnologias Utilizadas e Finalidade

- 1 Node.js – utilizado como ambiente de execução JavaScript no servidor.
- 2 Express.js – framework responsável por criar a API REST e gerenciar rotas HTTP.
- 3 Prisma ORM – ferramenta que abstrai o acesso ao banco de dados, facilitando a manipulação dos dados.
- 4 MySQL – banco de dados relacional utilizado para armazenar setores e patrimônios.
- 5 ExcelJS – biblioteca utilizada para geração de relatórios no formato Excel (.xlsx).
- 6 HTML5 – responsável pela estrutura semântica das páginas.
- 7 CSS3 – utilizado para estilização visual seguindo o padrão institucional do SENAI.
- 8 Bootstrap 5 – framework CSS para responsividade e componentes prontos.
- 9 JavaScript (ES Modules) – responsável pela interação do usuário e comunicação com a API.

4. Arquitetura API REST

A aplicação segue o padrão de arquitetura API REST. Neste modelo, o servidor não se preocupa com a interface gráfica, sendo responsável apenas por fornecer dados e regras de negócio através de endpoints bem definidos.

Os métodos HTTP utilizados possuem funções específicas:

- 1 GET – utilizado para consultar e listar informações.
- 2 POST – utilizado para cadastrar novos dados.
- 3 PUT – utilizado para atualizar informações existentes.
- 4 DELETE – utilizado para remover dados do sistema.

5. Estrutura do Back-end

A organização do back-end foi pensada para facilitar o entendimento dos alunos e a manutenção do sistema, separando responsabilidades de forma clara.

- 1 prisma/ – contém o schema do banco de dados e as migrations.
- 2 controllers/ – concentra as regras de negócio da aplicação.
- 3 routes/ – define os endpoints da API e seus métodos HTTP.
- 4 middlewares/ – realiza validações e tratamento de erros.
- 5 app.js – arquivo de configuração principal do Express.
- 6 server.js – responsável por iniciar o servidor.

6. Banco de Dados e Modelagem

O banco de dados foi modelado utilizando duas entidades principais: Setor e Patrimônio. Cada patrimônio está vinculado a um setor, caracterizando um relacionamento do tipo um-para-muitos.

Foi utilizado um enum para representar o status do patrimônio (ATIVO, MANUTENCAO e BAIXADO), garantindo padronização, integridade e melhor controle das informações.

7. Front-end e Experiência do Usuário

O front-end foi desenvolvido com foco na clareza visual e facilidade de uso. A interface permite que o usuário realize todas as operações de forma intuitiva, sem necessidade de conhecimentos técnicos.

As telas principais do sistema são:

- 1 Página Inicial – acesso rápido aos módulos do sistema.
- 2 Tela de Setores – cadastro, edição, exclusão e exportação para Excel.
- 3 Tela de Patrimônios – cadastro completo, filtros, status e exportação para Excel.

8. Funcionalidade de Exportação para Excel

A funcionalidade de exportação permite que os dados cadastrados sejam transformados em relatórios no formato Excel. Essa funcionalidade é amplamente utilizada em ambientes corporativos para análise, auditoria e controle patrimonial.

9. Como Configurar e Executar o Projeto

- 1 Instalar o Node.js (versão LTS recomendada).
- 2 Instalar o MySQL e criar o banco de dados.
- 3 Configurar o arquivo .env com a variável DATABASE_URL.
- 4 Executar npm install na pasta api.
- 5 Executar npx prisma migrate dev para criar as tabelas.
- 6 Iniciar o servidor com node src/server.js.
- 7 Abrir o front-end utilizando Live Server ou servidor estático.

10. Considerações Finais

Este projeto atende plenamente aos requisitos da avaliação prática do SENAI, demonstrando domínio técnico, organização, clareza didática e aplicação de conceitos fundamentais do desenvolvimento web moderno.