

# **Projeto: Sistema de Gerenciamento de Locadora**

## **Descrição do Projeto**

Desenvolver um sistema simples para gerenciar filmes de uma locadora. O sistema será composto por um back-end com uma API RESTful e um front-end utilizando Vue.js, Tailwind CSS e Atomic Design. O ambiente será configurado e executado com Docker e Docker Compose.

## **Funcionalidades do Sistema**

### **Back-End**

#### **1. API RESTful com Node.js:**

- CRUD de Filmes:
  - Criar um novo filme.
  - Listar filmes.
  - Atualizar informações de um filme.
  - Deletar um filme.
- Banco de Dados MySQL para armazenar os dados dos filmes.

#### **2. Modelo de Filme (Schema):**

- ID (gerado automaticamente).
- Título do filme.
- Gênero.
- Ano de lançamento.
- Status (disponível ou alugado).

### **Front-End**

## **Projeto: Sistema de Gerenciamento de Locadora**

### **1. Interface de Usuário com Vue.js:**

- Tela de listagem de filmes.
- Formulário para adicionar e editar filmes.
- Botão para excluir filmes.
- Indicador de status (disponível ou alugado).

### **2. Estilização com Tailwind CSS:**

- Layout responsivo.
- Uso de componentes estilizados.

### **3. Atomic Design:**

- Átomos: Botões, inputs e títulos.
- Moléculas: Cards de filmes e formulários.
- Organismos: Listagem de filmes com opções de ação.

## **Estrutura do Projeto**

### **1. Back-End:**

- Node.js com Express: API configurada para gerenciar os filmes.
- Prisma: Conexão com o banco MySQL.
- Banco de Dados: MySQL rodando em um container Docker.

### **2. Front-End:**

- Vue.js: Consumo da API do back-end para exibir e gerenciar filmes.
- Tailwind CSS: Estilização da interface.

## **Projeto: Sistema de Gerenciamento de Locadora**

- Atomic Design: Organização dos componentes.

### **3. Docker e Docker Compose:**

- Configurar os serviços:
  - Back-End: Aplicação Node.js.
  - Front-End: Aplicação Vue.js.
  - Banco de Dados: MySQL.
- Configuração de volumes e redes para comunicação entre os serviços.

## **Etapas para Desenvolvimento**

### **1. Configuração do Ambiente:**

- Instalar Docker e Docker Compose.
- Criar um arquivo docker-compose.yml para orquestrar os serviços:
  - Backend: Node.js.
  - Frontend: Vue.js.
  - Database: MySQL.

### **2. Back-End:**

- Criar uma API RESTful:
  - Instalar dependências: Express, Prisma, dotenv.
  - Configurar rotas para o CRUD.
- Configurar o Prisma:
  - Criar modelos e migrações para o banco MySQL.
- Criar um Dockerfile para o back-end.

## **Projeto: Sistema de Gerenciamento de Locadora**

### **3. Front-End:**

- Criar um projeto Vue.js:
  - Instalar dependências: Vue Router, Axios, Tailwind CSS.
  - Criar componentes baseados em Atomic Design.
- Configurar o consumo da API:
  - Fazer chamadas para as rotas do back-end (Axios).
- Criar um Dockerfile para o front-end.

### **4. Docker Compose:**

- Configurar o docker-compose.yml com:
  - Serviço para o back-end (Node.js).
  - Serviço para o front-end (Vue.js).
  - Serviço para o banco de dados (MySQL).
- Configurar volumes e portas para cada serviço.