

# **Relatório Técnico – Sistema de Gerenciamento Interno TechManage Solutions**

---

## **1. Introdução e Justificativa**

A **TechManage Solutions** é uma empresa voltada para o desenvolvimento de soluções tecnológicas. Atualmente, os registros de informações internas são realizados de forma manual, utilizando planilhas e documentos isolados.

Esse método tem causado **inconsistências nos dados, perda de informações e baixa produtividade**, prejudicando o controle de projetos e tarefas da organização.

Com o intuito de aprimorar o processo de gestão interna, propõe-se o desenvolvimento de uma **aplicação web automatizada**. Essa aplicação permitirá centralizar as informações da empresa em um sistema único, garantindo **organização, segurança e eficiência operacional**.

O sistema permitirá o **gerenciamento completo de usuários, projetos e tarefas**, substituindo os métodos manuais por um ambiente digital, moderno e escalável.

---

## **2. Objetivos do Sistema**

### **Objetivo Geral**

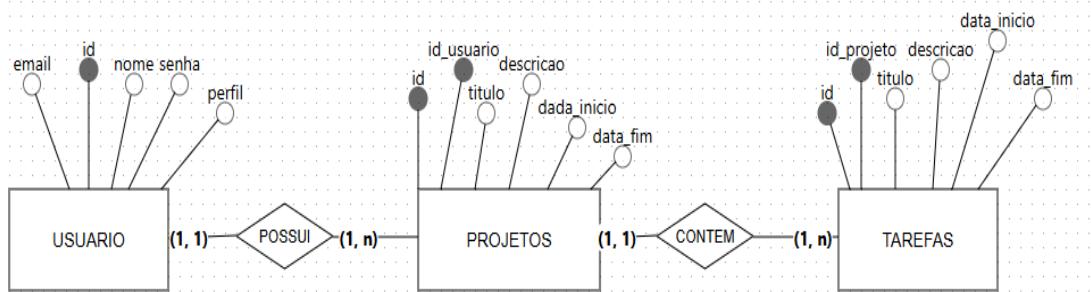
Desenvolver uma aplicação web CRUD completa para automatizar o controle interno da empresa TechManage Solutions.

### **Objetivos Específicos**

- Criar um sistema que permita **cadastrar, consultar, atualizar e excluir** registros de usuários, projetos e tarefas.
- Implementar **relacionamentos entre as entidades**, garantindo integridade referencial.
- Utilizar tecnologias modernas como **Node.js, Express.js, Sequelize e MySQL**.

- Garantir a **persistência e consistência dos dados** através de um banco relacional.
  - Fornecer uma **API REST** que retorne respostas JSON claras e estruturadas.
  - Aumentar a **produtividade** e reduzir erros operacionais no controle de informações internas.
- 

### 3. Modelagem do Sistema



#### Estrutura de Entidades

O sistema foi modelado com três entidades principais:

1. **Usuários (Users)**  
Representa os colaboradores cadastrados no sistema.
  2. **Projetos (Projects)**  
Representa os projetos desenvolvidos pelos usuários.
  3. **Tarefas (Tasks)**  
Representa as tarefas vinculadas a cada projeto.
- 

#### Relacionamentos

- **Usuário 1 : N Projetos** – Um usuário pode ter vários projetos.
  - **Projeto 1 : N Tarefas** – Um projeto pode conter várias tarefas.
- 

#### Modelo Conceitual (MER)

[USUÁRIOS] 1/N [PROJETOS] 1/N [TAREFAS]

---

## Modelo Lógico (DER)

Entidade	Atributos	Tipo de Chave
Users	id (PK), nome, email, senha, perfil	PK: id
Projects	id (PK), titulo, descricao, data_inicio, data_fim, id_usuario (FK → Users.id)	PK: id / FK: id_usuario
Tasks	id (PK), titulo, descricao, status, prioridade, id_projeto (FK → Projects.id)	PK: id / FK: id_projeto

## Descrição das Entidades

### Usuários

Atributo	Tipo	Descrição
id	INT (PK)	Identificador único do usuário
nome	VARCHAR(100)	Nome completo
email	VARCHAR(100)	E-mail do usuário
senha	VARCHAR(255)	Senha criptografada
perfil	ENUM('admin','colaborador')	Nível de acesso do usuário

### Projetos

Atributo	Tipo	Descrição
id	INT (PK)	Identificador único do projeto
titulo	VARCHAR(150)	Nome do projeto
descricao	TEXT	Descrição detalhada
data_inicio	DATE	Data de início do projeto
data_fim	DATE	Data de término previsto
id_usuario	INT (FK)	Usuário responsável pelo projeto

### Tarefas

Atributo	Tipo	Descrição
id	INT (PK)	Identificador único da tarefa
titulo	VARCHAR(150)	Nome da tarefa
descricao	TEXT	Detalhes sobre a atividade
status	ENUM('pendente','em_andamento','concluída')	Situação atual da tarefa
prioridade	ENUM('baixa','média','alta')	Grau de prioridade
id_projeto	INT (FK)	Projeto vinculado

## 4. Descrição Técnica do Sistema

### Tecnologias Utilizadas

Tecnologia	Finalidade
<b>Node.js (v18+)</b>	Plataforma JavaScript para o backend.
<b>Express.js</b>	Framework para criação de rotas e manipulação de requisições HTTP.
<b>Sequelize ORM</b>	Mapeamento objeto-relacional para integração com o banco de dados.
<b>MySQL</b>	Sistema Gerenciador de Banco de Dados relacional.
<b>dotenv</b>	Gerenciamento de variáveis de ambiente (.env).
<b>Nodemon</b>	Atualização automática do servidor em modo desenvolvimento.

### Estrutura de Pastas

```
project/
└── src/
    ├── config/
    │   └── database.js
    ├── models/
    │   ├── user.js
    │   ├── project.js
    │   └── task.js
    ├── controllers/
    │   ├── userController.js
    │   ├── projectController.js
    │   └── taskController.js
    ├── routes/
    │   ├── userRoutes.js
    │   ├── projectRoutes.js
    │   └── taskRoutes.js
    └── app.js
├── .env
└── package.json
README.md
```

### Variáveis de Ambiente (.env)

```
PORT=3000
DB_HOST=localhost
DB_USER=root
DB_PASS=senha
DB_NAME=techmanage_db
```

### Scripts do Projeto

```
"scripts": {
  "start": "node src/app.js",
  "dev": "nodemon src/app.js"
}
```

## 5. Funcionalidades CRUD

Cada entidade possui rotas específicas para as operações básicas do sistema.

Método	Rota	Descrição
<b>POST</b>	/users, /projects, /tasks	Cadastrar novo registro
<b>GET</b>	/users, /projects, /tasks	Listar todos os registros
<b>GET</b>	/users/:id, /projects/:id, /tasks/:id	Buscar por ID
<b>PUT</b>	/users/:id, /projects/:id, /tasks/:id	Atualizar registro
<b>DELETE</b>	/users/:id, /projects/:id, /tasks/:id	Excluir registro

Todas as respostas retornam mensagens em formato **JSON**, com tratamento de erros e validação dos dados.

---

## 6. Evidências de Execução

Durante a implementação e os testes do sistema, foram registradas as seguintes evidências:

- Execução do servidor Node.js (`npm run dev`).
  - Criação das tabelas no MySQL através das migrations do Sequelize.
  - Inserção de dados iniciais por meio dos seeds.
  - Testes das rotas via **Insomnia/Postman** para operações CRUD (POST, GET, PUT, DELETE).
  - Consultas SQL comprovando os relacionamentos entre as tabelas **Users**, **Projects** e **Tasks**.
- 

## 7. Conclusão

O desenvolvimento deste sistema proporcionou a criação de uma ferramenta eficiente para o **gerenciamento interno da TechManage Solutions**. A aplicação web, construída com tecnologias modernas, garante **centralização de dados, facilidade de acesso e aumento de produtividade**.

Além de cumprir os requisitos funcionais e técnicos exigidos, o projeto demonstrou a importância da **modelagem de dados** e do uso de **boas práticas de desenvolvimento backend**.

Com isso, a empresa passa a contar com um sistema robusto, escalável e seguro para o controle de suas informações internas.