# **Documento de Descrição da Solução ERP**

## **1. Visão Geral do Projeto**

### **1.1. Objetivo**

O objetivo deste projeto é desenvolver um sistema ERP completo e escalável que permita a gestão integrada de diferentes áreas de uma empresa, como vendas, estoque, finanças, e recursos humanos. O sistema será acessível via web e fornecerá uma interface moderna, segura e intuitiva.

### **1.2. Público-alvo**

Pequenas e médias empresas que necessitam de uma solução robusta para gerenciamento interno de operações, com a possibilidade de customização e escalabilidade conforme o crescimento da organização.

## **2. Arquitetura da Solução**

### **2.1. Arquitetura Geral**

A solução será composta por três camadas principais:

* **Frontend (ReactJS):** Responsável pela interface do usuário, com componentes modulares e reutilizáveis, que irão consumir dados da API do backend.
* **Backend (NestJS):** Implementará a lógica de negócio e exporá APIs REST ou GraphQL para o frontend e outros sistemas integrados.
* **Banco de Dados (MySQL):** Guardará dados estruturados, como informações sobre produtos, usuários, transações, entre outros.

### **2.2. Diagrama da Arquitetura**

Um diagrama que mostra o fluxo de comunicação entre o **frontend**, **backend** e **banco de dados**, além da integração com serviços externos.

scss

Copiar código

Cliente (Browser)

|

v

ReactJS (Frontend) <--> NestJS (Backend) <--> MySQL (Database)

|

v

APIs Externas/Integrações

## **3. Tecnologias Utilizadas**

### **3.1. Frontend (ReactJS)**

* **ReactJS**: Para desenvolvimento de interfaces dinâmicas e interativas.
* **Redux**: Para gerenciar o estado global da aplicação.
* **Axios**: Para fazer requisições HTTP ao backend.
* **Material-UI**: Para componentes de interface modernos e responsivos.
* **React Router**: Para roteamento interno das páginas da aplicação.

### **3.2. Backend (NestJS)**

* **NestJS**: Framework para desenvolvimento de APIs escaláveis e modulares.
* **TypeORM**: ORM para mapeamento de dados MySQL.
* **JWT (JSON Web Tokens)**: Para autenticação de usuários.
* **Swagger**: Para documentar as APIs REST e facilitar a integração.

### **3.3. Banco de Dados (MySQL)**

* **MySQL**: Banco de dados relacional para armazenar dados transacionais e informações da aplicação.

### **3.4. Infraestrutura**

* **Docker**: Para containerização e isolamento do ambiente de desenvolvimento e produção.
* **Nginx**: Para servir o frontend e proxy das requisições ao backend.
* **Kubernetes**: Para orquestrar múltiplos containers e escalar a aplicação de acordo com a demanda.

## **4. Funcionalidades do ERP**

### **4.1. Módulo de Vendas**

* Cadastro de produtos, clientes e vendedores.
* Gestão de cotações, pedidos e notas fiscais.
* Relatórios de vendas e comissão de vendedores.

### **4.2. Módulo de Estoque**

* Controle de entradas e saídas de estoque.
* Inventário em tempo real.
* Relatórios de movimentação de estoque.

### **4.3. Módulo Financeiro**

* Gestão de contas a pagar e receber.
* Controle de fluxo de caixa.
* Emissão de relatórios financeiros (lucro, despesas, balanço).

### **4.4. Módulo de Recursos Humanos**

* Cadastro de funcionários e cargos.
* Gestão de folha de pagamento e benefícios.
* Controle de ponto e banco de horas.

## **5. Segurança**

### **5.1. Autenticação e Autorização**

* Implementação de autenticação via **JWT** para controle de sessões de usuários.
* Autorização baseada em permissões (RBAC) para garantir que diferentes usuários (administradores, operadores, etc.) tenham acesso apenas às funcionalidades permitidas.

### **5.2. Criptografia**

* Todas as senhas de usuários serão criptografadas usando **bcrypt**.
* O tráfego entre frontend e backend será criptografado via **HTTPS**.

### **5.3. Monitoramento de Segurança**

* Implementação de monitoramento de logs e acessos utilizando ferramentas como **Sentry** ou **Datadog** para detectar atividades suspeitas.

## **6. Processos de Desenvolvimento**

### **6.1. Controle de Versão (Git)**

Todo o código será versionado usando o **Git**, e o repositório será hospedado no **GitHub** ou **GitLab**. Cada funcionalidade será desenvolvida em branches separadas e integradas via pull requests.

### **6.2. CI/CD (Integração e Deploy Contínuos)**

A pipeline de CI/CD será configurada com **GitHub Actions** ou **GitLab CI** para automatizar os testes, builds e deploys em ambientes de desenvolvimento, staging e produção.

### **6.3. Testes**

* **Testes Unitários:** Serão implementados usando **Jest** (para backend) e **React Testing Library** (para frontend).
* **Testes End-to-End:** Serão implementados com **Cypress** para garantir que todas as funcionalidades estejam funcionando corretamente de ponta a ponta.

## **7. Plano de Implantação**

### **7.1. Ambientes**

* **Desenvolvimento:** Utilização de containers Docker localmente.
* **Staging:** Ambiente para testes finais antes da produção, também containerizado.
* **Produção:** Deploy em um serviço de nuvem, como **AWS**, utilizando **Docker** e **Kubernetes** para orquestração dos serviços.

### **7.2. Migração de Dados**

Caso o cliente já possua dados legados (em planilhas ou outros sistemas), será implementado um processo de migração para importar esses dados para o MySQL.

## **8. Plano de Manutenção e Suporte**

### **8.1. Atualizações**

* Atualizações periódicas de segurança e melhorias de performance.
* Novas funcionalidades baseadas nas demandas dos clientes.

### **8.2. Monitoramento e Alertas**

* Implementação de monitoramento em tempo real com **Grafana** e **Prometheus** para acompanhar a performance do sistema.
* Configuração de alertas automáticos para falhas e indisponibilidades.

### **8.3. Suporte Técnico**

* Suporte será oferecido via e-mail, ticket e chat, com níveis de resposta de acordo com o plano contratado pelos clientes.

## **9. Conclusão**

Este documento oferece uma visão completa sobre o desenvolvimento da solução ERP, com as tecnologias escolhidas e as funcionalidades que serão implementadas. Com uma arquitetura modular e escalável, o sistema estará preparado para atender as demandas de empresas de pequeno e médio porte, com a possibilidade de crescimento e customizações conforme necessário.