



# TECHTINS

---

## SISTEMA DE GERENCIAMENTO DE USUÁRIOS E-COMMERCE DE TÊNIS

### 1. INTRODUÇÃO

A empresa *SneakerShop*, um e-commerce especializado na venda de tênis de diversas marcas, está em expansão e precisa de uma ferramenta interna para gerenciar os usuários que acessam o sistema, sendo eles administrador, cliente e funcionário. A administração da empresa requer um sistema para cadastro, visualização, edição e exclusão de usuários, com um histórico que permite acompanhar as últimas atividades de gerenciamento.

### 2. OBJETIVO

O objetivo do sistema é fornecer à *SneakerShop* uma aplicação que permita aos administradores gerenciar usuários que fazem parte da operação do e-commerce. O sistema será responsável por realizar o **CRUD de usuários** (Criar, Ler, Atualizar, Deletar), armazenando informações básicas de cada usuário.

A interface gráfica deve permitir que os administradores tenham acesso a uma lista de usuários e suas informações, além de um histórico recente das operações realizadas no sistema.

### 3. REQUISITOS FUNCIONAIS

#### 3.1. Funcionalidades para o administrador

O administrador deve ser capaz de:

- Visualizar uma lista de todos os usuários cadastrados;
- Cadastrar novos usuários no sistema;
- Editar informações de usuários existentes;
- Excluir usuários do sistema.



Como funcionalidade extra, o administrador pode:

- Visualizar um histórico dos últimos 5 usuários cadastrados, incluindo informações sobre quem fez o cadastro e quando.

### 3.2. Informações de cada usuário

O sistema deve gerenciar as seguintes informações mínimas de cada usuário:

- Nome Completo;
- E-mail;
- Função (Admin, Cliente, Funcionário);
- Data de Cadastro;
- Status (Ativo/Inativo).

### 3.3. Autenticação e Controle de Acesso

- O sistema deve permitir que os administradores façam login no sistema de forma segura, utilizando e-mail e senha;
- Apenas administradores logados podem acessar o sistema e gerenciar os usuários;
- A segurança das informações deve ser garantida, com todas as rotas protegidas e criptografia de senhas.

## 4. REQUISITOS NÃO FUNCIONAIS

- **Segurança:** O sistema deve garantir a criptografia das senhas dos usuários e proteger todas as rotas sensíveis com autenticação via token JWT;
- **Manutenção:** O código deve ser limpo, modularizado e seguir boas práticas de programação;
- **Usabilidade:** A interface deve ser simples, intuitiva e responsiva, permitindo fácil uso em dispositivos móveis e desktops;
- **Documentação:** O código e as funcionalidades devem ser documentados no GitHub, com instruções claras sobre instalação e execução de cada método.



## 5. DIVISÃO DAS TAREFAS POR STACK

### 5.1. Desenvolvedores back-end - Sistema de Segurança

Os **dois** desenvolvedores back-end devem desenvolver as seguintes funcionalidades:

- Implementar a lógica de autenticação utilizando JWT;
- Criar as rotas para o CRUD de usuários (criar, listar, editar, deletar) com proteção de acesso;
- Gerenciar permissões de usuários (diferenciar admins e usuários comuns);
- Criar um banco de dados relacional ou não relacional, (o que preferir), para armazenar informações dos usuários;
- Garantir a segurança das rotas com middleware de autenticação;
- Implementar um sistema de logging para registrar as operações feitas no sistema (cadastramento, edição e exclusão de usuários).

Como funcionalidade extra pode:

- Desenvolver a funcionalidade de histórico de usuários recentes, armazenando quem fez o cadastro e a data;
- Criar testes unitários e de integração para garantir o funcionamento adequado da API.

Os desenvolvedores devem documentar todas as rotas e funcionalidades da API para facilitar a integração.

### 5.2. Desenvolvedores front-end - Interface Gráfica

O desenvolvedor front-end devem desenvolver as seguintes funcionalidades:

- Desenvolver a interface utilizando algum framework de desenvolvimento, (escolha o que preferir), para criar um sistema visualmente intuitivo;
- Implementar uma página de login para que os administradores possam acessar o sistema de forma segura;



# TECHTINS

---

- Desenvolver as páginas de:
  - Lista de Usuários: Exibir todos os usuários cadastrados com a opção de edição e exclusão;
  - Cadastro de Usuário: Formulário para cadastrar novos usuários.

Como funcionalidade extra pode:

- Histórico de Usuários: Exibir os últimos 5 usuários cadastrados com informações sobre quem os registrou;
- Garantir que a interface seja responsiva e de fácil navegação, com feedback visual adequado (mensagens de sucesso e erro);
- Implementar a comunicação com a API de back-end para realizar as operações CRUD e exibir o histórico.

O desenvolvedor deve documentar todas as rotas e funcionalidades para facilitar a integração.

## 6. ENTREGA E DOCUMENTAÇÃO

Cada desenvolvedor deve postar sua respectiva parte no GitHub, com um README contendo:

- Descrição do sistema;
- Instruções para rodar a aplicação localmente (front-end e back-end);
- Descrição detalhada das rotas e funcionalidades implementadas;
- Exemplos de como testar as funcionalidades, podendo utilizar documentação, texto, vídeo, áudio, desenho, entre outros.

Os desenvolvedores back-end devem:

- Incluir uma documentação detalhada das rotas e da lógica utilizada;
- Descrever as rotas de CRUD e autenticação com exemplos de chamadas.

O desenvolvedor front-end devem:

- Incluir instruções de como configurar o ambiente e conectar com o backend;



# TECHTINS

---

- Descrever a estrutura de pastas e o processo de desenvolvimento da interface, e conexão entre as partes.

Todo o código e documentação devem ser organizados e versionados no GitHub.

## 7. CONSIDERAÇÕES FINAIS

Este projeto tem como foco fornecer uma ferramenta essencial para que a equipe de administração da **SneakerShop** gerencie os usuários do e-commerce de forma simples e eficaz. A aplicação deve estar bem documentada e preparada para ser apresentada ao cliente, explicando todas as funcionalidades desenvolvidas e como elas foram integradas para resolver as necessidades da empresa. Ao final, o sistema deve estar funcionando com todas as funcionalidades listadas e rodando localmente para testes e futuras implementações.