# Manual de Implementação da API de Autenticação

## 1. Tecnologias Utilizadas

* Node.js com TypeScript
* Express.js
* JWT (JSON Web Token)
* Bcrypt para hash de senha
* Banco de dados relacional (ex: PostgreSQL ou MySQL, no projeto estou usando sqlite)

## 2. Estrutura de Diretórios (Exemplo)

* /src
* ├ controllers
* ├middlewares
* ├ routes
* ├services
* ├models

## 3. Endpoints da API

* POST /usuario - Cadastro de novo usuário
* POST /entrarusuario - Autenticação e geração de token do usuário
* POST /entrarAdm - Autenticação e geração de token do administrador
* GET /usuario - Retorna dados do usuário autenticado
* GET /logs - Retorna registros de ações (acesso restrito ao admin)

## 4. Lógica de Autenticação

* 1. Usuário envia e-mail e senha para /login.
* 2. API valida a senha com bcrypt.
* 3. API gera token JWT com ID, e-mail e role.
* 4. Front-end armazena o token.
* 5. Requisições protegidas enviam o token via header Authorization.

## 5. Segurança e LGPD

* Senhas nunca são armazenadas em texto puro.
* Tokens têm tempo de expiração.
* Somente usuários autenticados acessam suas próprias informações.
* Logs de ações são protegidos e acessíveis apenas a administradores.

## 6. Execução

* 1. Instale as dependências com `yarn`.
* 2. Configure as variáveis de ambiente (.env).
* 3. Rode a aplicação com `yarn dev`.
* 4. A API estará disponível em `http://localhost:3333` (por padrão).
* 5. O front-end está em um arquivo a parte e roda por padrão na porta `http://localhost:5500` (por padrão).