Especificação da API de Autenticação em Node.js

Objetivo

Este documento descreve os requisitos e especificações para o desenvolvimento de uma API de autenticação em Node.js. A API será responsável por autenticar usuários com base em suas credenciais (e-mail/senha), comparando a senha fornecida com o hash armazenado no banco de dados, e gerar um token de acesso JWT para autorização em rotas protegidas.

Requisitos Funcionais

Autenticação de Usuário:

- A API deve permitir que os usuários se autentiquem fornecendo seu e-mail e senha.
- A senha fornecida pelo usuário deve ser comparada com o hash armazenado no banco de dados para verificar a autenticidade.
- Se as credenciais forem válidas, a API deve gerar e retornar um token de acesso JWT.

Tecnologias Utilizadas

- Node.js: Plataforma de execução JavaScript para construção do servidor.
- Express.js: Framework web para Node.js, utilizado para criar APIs RESTful.
- PostgreSQL: Sistema de gerenciamento de banco de dados relacional para armazenar dados de usuários e tokens JWT.
- bcrypt.js: Biblioteca para hash de senhas de forma segura.
- jsonwebtoken: Biblioteca para geração e verificação de tokens JWT.

Fluxo de Funcionamento

1.O cliente faz uma solicitação de autenticação à API, fornecendo seu e-mail e senha.
2.A API verifica se o e-mail fornecido corresponde a um usuário cadastrado.
3.Se o e-mail existir, a API compara a senha fornecida com o hash da senha armazenada no banco de dados.
4.Se as credenciais forem válidas, a API gera um token JWT e o retorna para o cliente.
5.O cliente usa o token JWT em solicitações posteriores às rotas protegidas. As rotas protegidas verificam a validade do token e concedem acesso apenas a usuários autenticados.

Endpoints da API

- POST /api/auth/login: Endpoint para autenticar um usuário e obter um token JWT.