1. Objetivo do Projeto

O objetivo principal é desenvolver e implementar um Módulo de Autenticação (Login e Cadastro) seguro e funcional para o sistema **ROTA CERTA**. A solução utiliza Node.js (Express) para o backend, MySQL para a persistência de dados e o módulo crypto para garantir o armazenamento seguro das senhas via função hash SHA-256.

2. Escopo do Projeto

O escopo do projeto abrange:

- **Registro de Usuário:** Permitir o cadastro de novos usuários com nome, e-mail, data de nascimento e senha.
- **Armazenamento Seguro:** Garantir que todas as senhas sejam armazenadas como hash (SHA-256).
- **Autenticação:** Permitir que usuários cadastrados façam login validando a senha fornecida contra o hash armazenado.
- **Validações:** Incluir validações essenciais no backend para integridade dos dados (tamanho mínimo de senha, unicidade do e-mail).
- **Comunicação:** Disponibilizar rotas REST (/cadastro e /login) para comunicação com o frontend.

Os seguintes itens estão **fora do escopo** desta entrega:

- Recuperação de Senha ("Esqueci Minha Senha").
- Autorização, gestão de perfis ou níveis de acesso (Admin, Cliente, etc.).
- Implementação de Tokens de Sessão (como JWT ou Cookies) para manter o estado do usuário após o login.

3. Descrição Funcional das Rotas de Autenticação

Cadastro de Usuário (/cadastro)

A rota de cadastro é responsável por receber os dados de um novo usuário via requisição POST. Antes de salvar as informações, o sistema executa uma série de validações no backend para garantir a integridade e segurança dos dados.

Validações e Processamento: O sistema primeiramente verifica se o **e-mail** fornecido já existe no banco de dados, retornando um erro 409 (Conflito) caso positivo. Também são feitas verificações de preenchimento e formato: o campo **senha** deve ter no mínimo 6 caracteres e não pode estar vazio, enquanto os campos **e-mail** (deve conter '@') e **nome** (mínimo de 3 caracteres) também são validados.

Qualquer falha nessas regras resulta em um erro 400 (Requisição Inválida). Após a aprovação das validações, a senha fornecida é imediatamente convertida para um *hash* seguro (SHA-256) antes de ser persistida no banco de dados. Em caso de sucesso na inserção, o sistema retorna o código 200 (OK) com uma mensagem de confirmação de cadastro.

Autenticação de Usuário (/login)

A rota de login é responsável por autenticar o usuário, recebendo o e-mail e a senha via requisição POST. O sistema garante que os campos **e-mail** e **senha** sigam as regras básicas de validação (mínimo de 6 caracteres para a senha, formato válido para o e-mail).

Processamento e Retorno: O sistema consulta o banco de dados usando o e-mail. A senha fornecida na requisição é convertida para *hash* (SHA-256) para que possa ser comparada com o *hash* da senha armazenada.

- **Sucesso:** Se o *hash* fornecido for idêntico ao *hash* salvo, o sistema retorna o código 200 (OK) e os dados básicos do usuário (ID, nome e e-mail).
- **Falha:** Se o usuário não for encontrado ou se os *hashes* não coincidirem, o sistema retorna o código 401 (Não Autorizado) com a mensagem de "Usuário ou senha inválida".

Gerenciamento de Servidor e Erros

O servidor está configurado para usar o **protocolo CORS** para aceitar requisições do frontend. Além disso, as rotas de autenticação **restringem o acesso GET**, retornando 405 (Método Não Permitido) em vez de processar a requisição. O servidor inclui um **tratamento de erros robusto** (try...catch) para capturar falhas na conexão com o banco de dados ou erros internos, retornando o código 500 (Erro Interno do Servidor) ao frontend e registrando a exceção no console para análise.