

Projeto Fictício: Sistema de Chat em Tempo Real

Tecnologias Utilizadas:

BackEnd:

- Node.js;
- Express, MSSQL e Socket.io;
- SQL Server.

FrontEnd:

- Next.js;
- TypeScript;
- HTML/CSS;
- Socket.io.

Ambiente:

- Docker;
- GIT/GitHub.

Estrutura de Pastas:

- - chat-app/
 - backend/
 - src/
 - controllers/
 - models/
 - routes/
 - sockets/
 - app.js
 - server.js
 - frontend/
 - public/
 - src/
 - contexts/
 - components/
 - app/
 - App.tsx
 - next.config.js

Funcionalidades Mínimas do Projeto:

1. Autenticação local de usuário no banco de dados;
2. Envio e recebimento de mensagens privadas entre dois usuários;
3. Navegação entre múltiplas conversas;
4. Armazenamento das mensagens em banco de dados;
5. Atualização em tempo real das mensagens nos chats usando Socket.io.

Etapas de Desenvolvimento:

Configuração Inicial:

- Configurar a estrutura de pastas básica para o BackEnd e o FrontEnd.
- Iniciar repositório no GitHub.

BackEnd:

- Configurar um servidor Node.js com Express.
- Definir rotas e controladores para autenticação, conversas e mensagens.
- Integrar o SQL Server para armazenamento de dados.
- Implementar a comunicação via Socket.io para suporte a chat em tempo real.
- Configurar o ambiente Docker para empacotar a aplicação BackEnd.

FrontEnd:

- Iniciar um projeto Next.js e TypeScript.
- Criar páginas e componentes para autenticação, listagem de conversas e mensagens.
- Implementar a comunicação com a API Rest do BackEnd.
- Configurar a comunicação em tempo real usando Socket.io.
- Criar estilos CSS para uma interface de usuário amigável e responsiva.
- Configurar o ambiente Docker para empacotar a aplicação FrontEnd;

Usabilidade UX/UI:

- Criar um design agradável e intuitivo para a interface de usuário.
- Recomendamos que a interface seja responsiva em diferentes dispositivos.

Integração e Testes:

- Integrar o FrontEnd com o BackEnd para criar uma aplicação funcional.
- Testar a troca de mensagens entre usuários em tempo real usando Socket.io.
- Testar a usabilidade e fazer ajustes necessários.
- Testar performance da aplicação (recomendamos o uso do Lighthouse do Chrome para o frontend).

Mockup App:

Visão UX recomendada da aplicação:

