

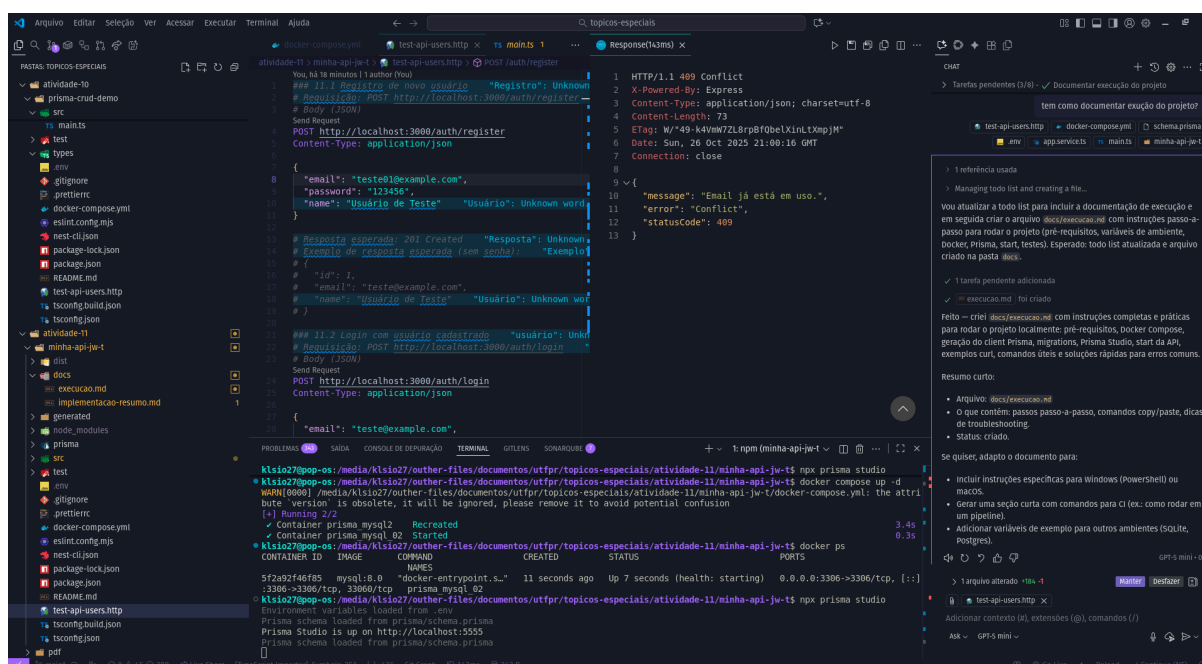


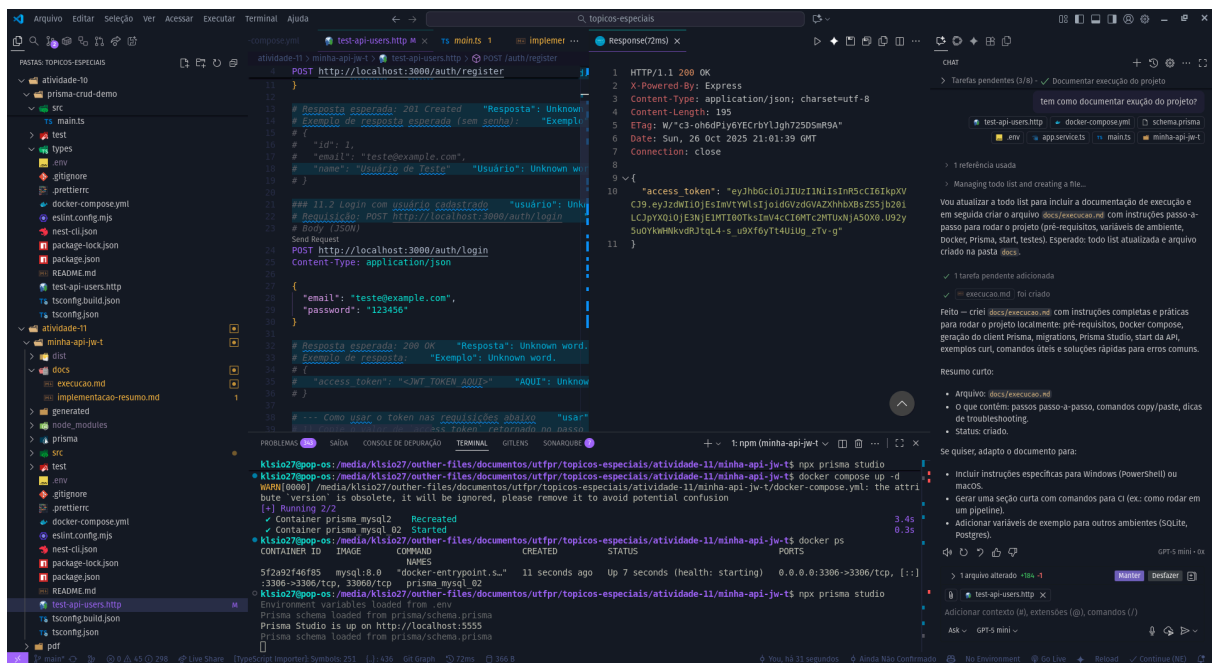
Atividade 11 | Autenticação com JWT e Prisma

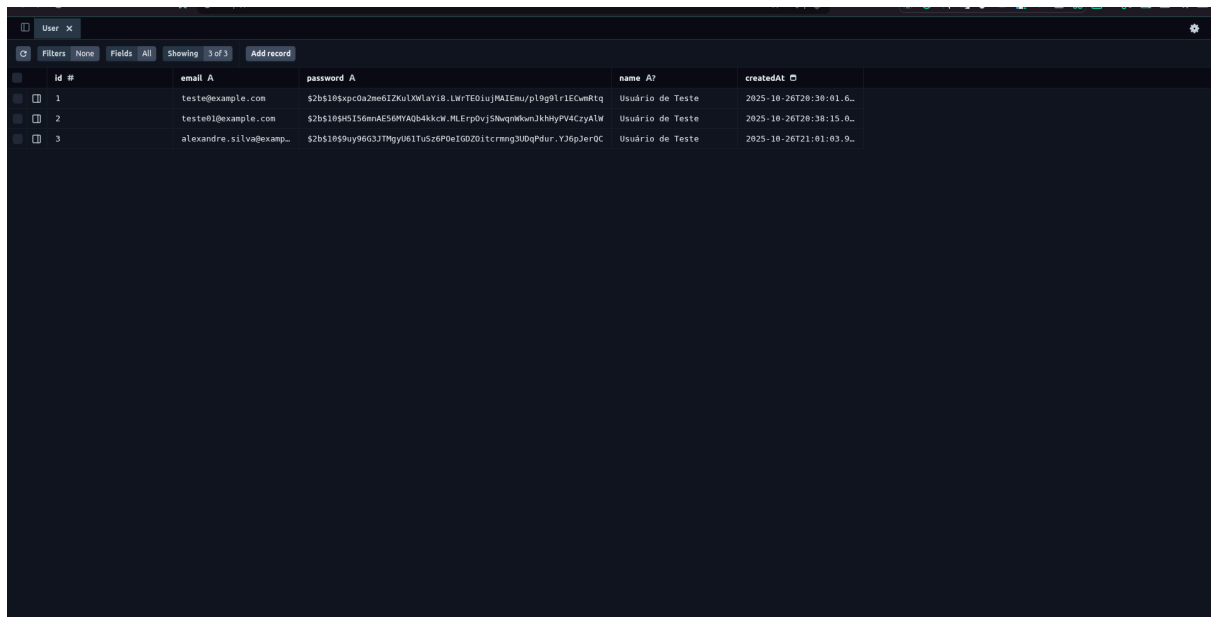
Evidências — Autenticação JWT (Atividade 11)

Capturas de tela

Insira aqui as capturas obtidas durante os testes (ex.: [docs/images/register.png](#) , [docs/images/login.png](#) , [docs/images/perfil-401.png](#) , [docs/images/perfil-200.png](#)).







id #	email A	password A	name A?	createdAt
1	teste@example.com	\$2b\$10\$xc0a2me6IZkLXWlAYi8.LWTEQiuJMAIEmu/pl8g9lr1ECwRtq	Usuário de Teste	2025-10-26T20:30:01.6...
2	teste01@example.com	\$2b\$10\$H5156nnAES6MYAQ84kckW.MLErPdvj5WqmqWsanJkHhYPV4CzYALW	Usuário de Teste	2025-10-26T20:30:15.6...
3	alexandre.silva@examp...	\$2b\$10\$9uy96G3JTMgyU6ITuS26PDeIG0Z0itcrmg3UDqPdur.Y36pJerQC	Usuário de Teste	2025-10-26T21:01:03.9...

Testes executados

Teste: Registro de novo usuário (POST /auth/register)

- Objetivo: Criar um usuário.
- Entrada: `{ "email": "alexandre.silva@example.com", "password": "123456", "name": "Usuário de Teste" }`
- Saída esperada: Objeto do usuário sem o campo `password`.
- Resultado: 201 Created

Teste: Login (POST /auth/login)

- Objetivo: Autenticar usuário e obter JWT.
- Entrada: `{ "email": "alexandre.silva@example.com", "password": "123456" }`
- Saída esperada: `{ "access_token": "<JWT>" }`
- Resultado: 200 OK

Teste: Acesso protegido sem token (GET /auth/perfil)

- Objetivo: Verificar proteção da rota.
- Entrada: sem header Authorization
- Saída esperada: 401 Unauthorized
- Resultado: 401 Unauthorized

Teste: Acesso protegido com token (GET /auth/perfil)

- Objetivo: Acessar rota protegida usando token válido.
 - Entrada: Header `Authorization: Bearer <JWT>`
 - Saída esperada: 200 OK com objeto `user` (sem senha)
 - Resultado: 200 OK
-

Perguntas e Respostas

1. O que o método `validate()` da `JwtStrategy` retorna e por que ele é importante?
 - Retorna o objeto do usuário (ou um payload seguro) correspondente ao `sub` do token; esse valor é atribuído a `request.user`. É importante porque identifica o usuário autenticado e permite autorizar/usar seus dados nas rotas protegidas.
 2. Qual o papel do `JwtAuthGuard` e o que acontece se ele não for usado em uma rota?
 - O `JwtAuthGuard` verifica a presença e validade do token JWT (via `JwtStrategy`). Se não for usado, a rota fica pública e qualquer cliente pode acessá-la sem autenticação; dados sensíveis não ficarão protegidos.
 3. O que o método `login()` do `AuthService` retorna? Por quê?
 - Retorna um objeto com `access_token` (o JWT assinado). Isso permite ao cliente enviar o token nas próximas requisições para provar sua identidade sem reautenticar a cada chamada.
-

FAQ

1. O conteúdo de um JWT pode ser lido por qualquer pessoa? Por que ainda assim ele é considerado seguro?
 - Sim: o payload é codificado em base64 e pode ser lido, mas não pode ser alterado sem invalidar a assinatura. A segurança vem da assinatura (secret ou chave privada) que garante integridade/autenticidade.
2. O que significa autenticação stateless e como ela funciona com JWT?
 - Stateless: o servidor não mantém sessão armazenada; toda a informação necessária vem no token (payload). Com JWT, o servidor

valida a assinatura do token e usa o payload para identificar o usuário sem consultar um store de sessão.

3. O JWT funciona com cookies? É seguro usar JWT dessa forma?

- Sim, pode ser usado em cookies. Segurança depende do uso: cookies HttpOnly + Secure + SameSite oferecem proteção contra XSS/CSRF; armazenar JWT em localStorage é mais suscetível a XSS. Avalie o fluxo e riscos antes de escolher.

Link para repositório:

<https://github.com/klsio22/topicos-especiais/tree/main/atividade-11/minha-api-jw-t>