INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO NORTE

INFORMÁTICA PARA INTERNET

EMILY MEDEIROS ALVES DOS SANTOS, KARYNE OHARA FREITA DE MEDEIROS, MATEUS LUCENA PEREIRA, MICKAELLE KARINE SOUZA SILVA, RIVERSSON PAULO DOS SANTOS

SESSÃO UTILIZANDO API REST ATRAVÉS DO FASTAPI

ATIVIDADE DE PESQUISA - STOCKFIELD

CAICÓ/RN
2025

1. Como enviar usuário e senha de forma segura?

- A melhor maneira de enviar credenciais (usuário e senha) de forma segura é utilizando HTTPS para criptografar a comunicação entre o cliente e o servidor.
- **Payload JSON:** Enviar os dados em formato JSON;
- **Hash de Senha:** Armazenar a senha de forma segura no banco de dados utilizando um algoritmo de hash, como bcrypt. Ao validar a senha, você deve comparar o hash gerado com o hash armazenado.

2. Como manter a sessão: no frontend ou backend?

É comum manter a sessão no backend, utilizando tokens. O fluxo típico é:

- **Login:** O usuário envia as credenciais e, se forem válidas, o servidor gera um token (como JWT - JSON Web Token) e o retorna ao cliente.
- **Armazenamento do Token:** O frontend deve armazenar o token, geralmente no localStorage ou sessionStorage, e incluí-lo em requisições subsequentes, como no cabeçalho Authorization.

3. Precisa de outras bibliotecas para o processo?

Sim, algumas bibliotecas podem facilitar o processo:

- **FastAPI:** Para a criação da API REST.
- **Passlib:** Para hashing de senhas.
- **PyJWT:** Para geração e validação de tokens JWT.
- **SQLAlchemy:** Para interação com o banco de dados.

4. Como você está implementando em seu projeto?

Ainda não implementamos, mas o fluxo de login/sessão será da seguinte forma:

1. **Cadastro de usuário**
    - Armazeno o **hash da senha** no banco (não a senha real).
2. **Login**
    - Usuário envia {username, password}.
    - Backend verifica o hash.

- Se correto → gera **JWT** com payload (user_id, exp, roles).

3. **Manutenção da sessão**
   - O frontend guarda o token (ex.: localStorage ou cookie seguro HttpOnly).
   - A cada requisição de CRUD no estoque (adicionar, listar, remover itens), o token é enviado no header.
   - O backend valida o token antes de acessar o banco.
4. **Renovação de sessão**
   - O JWT pode expirar em, por exemplo, **15 min**.
   - Para melhorar usabilidade, você pode usar **refresh token** (token de longa duração para pedir um novo access token).
5. **Segurança extra**
   - Ativar HTTPS obrigatório.
   - Se usar cookies → marcar como HttpOnly, Secure, SameSite=strict.
   - Se usar localStorage → cuidado com ataques XSS, sanitize inputs no frontend.