No desenvolvimento desse projeto com Flask, eu aprendi como funciona a criação de **múltiplas páginas em uma aplicação web** e como o servidor lida com a navegação entre elas.

Primeiro, entendi a importância das **rotas** (@app.route) no Flask. Cada rota representa um caminho da URL e está ligada a uma função que retorna algum conteúdo. Por exemplo, quando eu acesso "/" sem nada depois, o Flask executa a função associada e retorna a index.html que é a página inicial, e quando eu acesso "/sobre", ele retorna a página "Sobre". Isso me mostrou que o Flask funciona como um **controlador que decide qual resposta enviar para cada endereço acessado**.

Outra coisa que aprendi foi o uso de **templates HTML** com o render_template(). Em vez de escrever o HTML diretamente no código Python, eu criei arquivos .html dentro da pasta templates. Assim, o Flask busca esses arquivos e os exibe para o usuário. Isso é importante porque separa a parte visual (HTML) da parte lógica (Python), deixando o projeto mais organizado que futuramente será muito útil para organização dos conteúdos do RPG.

Também aprendi que a mudança de páginas acontece porque o navegador envia uma **requisição HTTP** ao servidor cada vez que eu clico em um link ou acesso uma rota. O Flask então identifica qual rota foi chamada e responde com o HTML correspondente. Isso significa que a navegação entre páginas não é apenas "trocar de arquivo", mas sim uma comunicação entre **cliente (navegador)** e **servidor (Flask)**.

Outro ponto importante foi entender o uso do url_for(). Ele serve para gerar automaticamente os links corretos entre as páginas, sem que eu precise escrever manualmente os caminhos das rotas. Isso evita erros e facilita caso eu altere alguma rota no futuro.

Resumindo, o que eu aprendi de mais importante nesse projeto foi:

- Como criar e organizar rotas no Flask.
- A diferença entre lógica do servidor (Python) e apresentação (HTML).
- Como funciona a comunicação cliente-servidor nas mudanças de página.
- O papel do render_template() para carregar arquivos HTML.
- A importância do url_for() para criar links dinâmicos e seguros.

Já usei algumas vezes o Flask, mas essa tarefa me ajudou a entender mais detalhadamente esse processo e entender melhor como aplicações web são estruturadas e como a navegação entre páginas acontece de forma controlada pelo servidor.