LAB 2 - Reuso de Software

1. Identificar onde reusar:

   API no Back-end

   A API será responsável por realizar a integração entre o banco de dados e o front-end do sistema. Por meio dela, serão reutilizadas estruturas já consolidadas de comunicação, bem como padrões de requisição, garantindo maior organização, segurança e padronização.

   Essa API será utilizada por todo o sistema, incluindo funcionalidades como **cadastro, login e operações internas**, uma vez que toda a comunicação entre o sistema e o banco de dados será realizada exclusivamente por meio dela.

   Framework no Front-end

   O framework definirá a estrutura e a organização da interface do aplicativo, permitindo a reutilização de componentes visuais e facilitando a manutenção do código.

   Ele será utilizado na construção de todas as páginas do sistema, garantindo consistência visual, melhor desempenho e maior escalabilidade da aplicação.

   Bibliotecas

   As bibliotecas serão empregadas para facilitar a manipulação de dados, a implementação das regras de negócio e o desenvolvimento de outras funcionalidades no back-end.

   Seu uso contribui para maior produtividade, redução de código repetitivo e aumento da confiabilidade do sistema.

2. Identificar artefatos de reuso (frameworks, componentes, bibliotecas, APIs)

   - API: FastAPI
   - FRAMEWORK: React
   - BIBLIOTECAS: SQLAlchemy, psycopg, math

3. Analisar critérios técnicos e de arquitetura de cada artefato reusado

- FastAPI:
    - Critérios técnicos: Alta performance, escrita em Python, fácil integração com banco de dados e criação de APIs. Além de ser considerado uma API madura e amplamente adotada na comunidade Python e em constante evolução.
    - Critérios de arquitetura: Baseada em APIs REST, fácil comunicação entre front-end e back-end e organizado em camadas do sistema.
- React:
    - Critérios técnicos: Framework JavaScript atualizado, reutilização de componentes, boa performance e comunidade de suporte.
    - Critérios de arquitetura: Baseado em componentes reutilizáveis, fácil manutenção e organização modular da interface.
- SQLAlchemy:
    - Critérios técnicos: Biblioteca ORM para Python, escrita de comando SQL reduzida e suporte a múltiplos bancos de dados com boa documentação.
    - Critérios de arquitetura: Melhor organização da estrutura do sistema pela separação da lógica de negócio da manipulação do banco.
- psycopg:
    - Critérios técnicos: Driver oficial e amplamente usado para conexão com PostgreSQL, além de boa documentação.
    - Critérios de arquitetura: Comunicação direta com banco de dados, garantindo integração segura com back-end.
- math:
    - Critérios técnicos: Biblioteca do Python para cálculos matemáticos, sendo estável e amplamente utilizada.
    - Critérios de arquitetura: Facilita cálculos internos do sistema com baixo impacto na arquitetura.

4. Analisar benefícios e riscos de reuso de seus artefatos

O reuso de artefatos facilita e acelera o desenvolvimento, reduzindo erros e utilizando de soluções previamente testadas para aumentar a produtividade e qualidade do sistema. Os frameworks e bibliotecas como FastAPI, React, SQLAchemy e psycopg possibilitam essa organização arquitetural melhorada e uma melhor separação de responsabilidades com foco na regra de negócio.

Porém, pode gerar dependências de tecnologias externas e possíveis problemas de compatibilidade com o projeto, uma vez que a arquitetura do sistema passa a ser influenciada pelos ideais dessas ferramentas e, como elas têm manutenções esporádicas, essas mudanças de versão ou perda de suporte podem gerar esses problemas de compatibilidade.