Resumo de Vídeo: Como criar e desacoplar camadas utilizando SOLID - Rodrigo Branas

O vídeo fornece um exemplo prático de como aplicar os princípios SOLID, especificamente o princípio de Inversão de Dependência (Dependency Inversion Principle - DIP) e o princípio Open/Closed, no desenvolvimento e estruturação de camadas de software. Abaixo, descrevo como esses conceitos foram incorporados no código demonstrado no vídeo:

Inversão de Dependência (DIP):

O youtuber inicia a codificação definindo uma estrutura onde as camadas de aplicação não dependem diretamente das implementações concretas, mas sim de abstrações. Por exemplo, ele destaca que camadas são definidas pelo relacionamento entre os componentes, não simplesmente pela organização em pastas, enfatizando que uma camada é delineada pelo quem conhece quem e quem se comunica com quem.

No vídeo, é demonstrado o início da criação de uma aplicação onde as camadas estão inicialmente misturadas, para depois refletir sobre como separá-las adequadamente. O código inicial não segue o DIP, mostrando a criação direta de instâncias e a alta dependência entre as camadas, preparando o cenário para uma refatoração que aplique o DIP.

Princípio Open/Closed:

Ao longo do live coding, o youtuber discute como cada camada deve ser aberta para extensão, mas fechada para modificação. Por exemplo, ao estruturar o código para criar e gerenciar transações financeiras, ele indica que as mudanças futuras no tratamento de transações não deveriam modificar as classes existentes, mas estender suas funcionalidades.

Este princípio é ilustrado na prática quando ele reflete sobre a necessidade de evitar que mudanças em uma camada específica causem a necessidade de alterações em outras camadas, mostrando

como o design adequado pode promover um sistema mais robusto e menos propenso a erros durante manutenções ou upgrades.

Aplicação Prática dos Princípios:

Durante o vídeo, o youtuber demonstra como começar uma aplicação com uma arquitetura simples e depois evoluir essa arquitetura aplicando os princípios SOLID. Ele começa com uma aplicação onde o acoplamento é alto e, passo a passo, discute como esse acoplamento pode ser reduzido.

Por exemplo, ele começa criando uma rota no Express que manipula diretamente os dados de entrada e interage com o banco de dados. Ao longo da sessão, ele sugere como isso pode ser refatorado para que a lógica de negócios seja isolada da camada de apresentação, cumprindo o DIP e preparando o terreno para aplicar o princípio Open/Closed.

Testes e Refatoração:

O youtuber também enfatiza a importância dos testes automatizados, mostrando como os testes podem guiar o desenvolvimento (Test Driven Development - TDD). Ele escreve testes que falham inicialmente e, em seguida, ajusta o código para passar nos testes, uma prática que garante que a aplicação se mantenha alinhada aos requisitos e princípios de design.

Este método não apenas aplica os princípios SOLID mas também garante que a aplicação possa evoluir de maneira controlada e previsível.

Este resumo demonstra como o vídeo utilizou exemplos práticos para explicar e aplicar os conceitos SOLID, focando em como eles podem melhorar significativamente a estruturação e manutenção das camadas de uma aplicação de software.