

Basicamente o vídeo demonstra por código como aplicar os princípios de SOLID que em resumo são:

Responsabilidade Única: Cada parte do programa deve fazer apenas uma coisa. Por exemplo, a parte que lida com os dados não deve se preocupar com a parte que mostra eles na tela.

Aberto/Fechado: Queremos que nosso programa possa crescer sem precisar mexer no que já está funcionando. Se precisarmos adicionar uma nova funcionalidade, devemos ser capazes de fazer isso sem mexer nas partes antigas.

Substituição de Liskov: As partes do programa devem poder ser trocadas umas pelas outras sem causar problemas. Por exemplo, se temos diferentes formas de armazenar dados, devemos poder trocar entre elas sem quebrar nada.

Segregação de Interface: Cada parte do programa deve ter acesso apenas ao que precisa. Por exemplo, se temos diferentes tipos de usuários, cada um deve ver só o que é importante para ele.

Inversão de Dependência: As partes do programa não devem depender diretamente umas das outras. Em vez disso, devem depender de algo que as conecte. Por exemplo, em vez de uma parte do programa chamar diretamente outra, ela deve usar algo como uma mensagem para se comunicar.

Também é abordado princípios de Clean Architecture que basicamente é um padrão de arquitetura de software que visa criar sistemas robustos, escaláveis e testáveis, com uma estrutura clara e independente de detalhes de implementação. A ideia central da Clean Architecture é separar o código em camadas bem definidas e altamente coesas, onde cada camada possui responsabilidades específicas e dependências direcionadas para dentro.