

**Disciplina de ANÁLISE E PROJETO DE SISTEMAS 2**  
**Arquitetura Limpa - Aulas de 24/11/2023**  
**Aluno: Guilherme Baumgratz Spiger**  
**RA: 2460823**

O vídeo Clean Architecture - Imersão FullStack com Rodrigo Branas ([disponível aqui](#)), apresenta conceitos modernos de arquitetura de Software, que foram desenvolvidos e apresentados ao público a partir do início dos anos 2000, pelas contribuições de diversas pessoas influentes no meio de desenvolvimento de Software. A apresentação foca em aspectos mais práticos desses conceitos, deixando um pouco de lado o rigor acadêmico, o que é compreensível dado a intenção aparente da apresentação: Introduzir a importância de desenvolvedores dedicarem mais tempo a definição da arquitetura do software a ser desenvolvido.

Sobre o conteúdo apresentado, fica claro que, inicialmente, um desenvolvedor ou uma equipe de desenvolvimento pode não ter interesse em desenvolver o software com a arquitetura em mente e apenas escrever um código que funcione e resolva o problema, com a intenção de entregar resultados no menor tempo possível. Isso realmente faz com que o desenvolvimento seja mais produtivo (Horas de Desenvolvimento/Software Entregue), no entanto isso tende a produzir software sem cuidado com a sua manutenção de longo prazo, o que pode dificultar a alteração de regras do negócio, inclusão de novas funcionalidades ou mudanças de tecnologias, exigindo que sejam gastas muitas e muitas horas de desenvolvimento, reescrevendo código, e consequentemente reduzindo a produtividade.

O principal motivo, apresentado no vídeo, da causa dessas dificuldades de manutenção e de escalabilidade, é o alto nível de acoplamento no software, que pode ser visualizado através da instanciação de classes diretamente em outras classes, e indica que em uma mudança de uma classe reflete em alterar também todos os locais onde está é instanciada. Uma solução apresentada no vídeo é segregar o software em camadas que agreguem responsabilidades únicas e usar de técnicas como a injeção de dependências, interfaces e DTOs (data transfer objects) para fazer a comunicação entre essas camadas. O intuito é fazer com que as alterações realizadas em uma camada não forcem alterações nas outras camadas.

Finalmente, os apresentadores comentam que não existe uma fórmula pronta a ser seguida e deve-se ter o bom senso para adaptar conforme a situação. Também indicam que produzir software utilizando esses conceitos de arquitetura é o ideal para novos projetos e que pode ser adotado também, de maneira gradual reescrevendo partes ou módulos, para softwares legados, mas que não é algo obrigatório e pode consumir tempo extra de desenvolvimento de uma aplicação pequena que não aproveitará muito dos benefícios de escalabilidade que uma arquitetura “Clean” proporciona.