O que é clean code

Introdução

Clean code é uma filosofia de desenvolvimento de software, não é uma metodologia rígida (Um conjunto de regras fixas ou obrigatórias) e sim uma abordagem, uma forma de tornar os códigos mais entendíveis pelos seres humanos por meio de boas práticas que podem ser adaptadas por cada pessoa ou equipe. Por fim, o clean code busca tornar os códigos mais claros e intuitivos.

É importante entender que por não ser um conjunto de regras absolutas o Clean Code não é algo decorável, é experiencial. Clean Code exige prática, revisão e adaptação constante ao longo do tempo, ou seja, Clean Code é um conceito que é construído ao longo das experiências e para que um programador comece a escrever “códigos limpos" é necessário que ele escreva muitos “códigos sujos” anteriormente.

Os 3 Pilares do Clean Code

Clean Code é pautado em 3 fundamentos que representam objetivos que um código deve alcançar para ser considerado um código limpo. Focando na qualidade em vez da velocidade para garantir que o código seja fácil de entender e que mantenha a qualidade a longo prazo, pois um código bem estruturado é um código que pode ser lido, mantido e expandido.

Os 3 pilares:

Legibilidade

O principal objetivo é tornar as tarefas de leitura e escrita de códigos mais fáceis e intuitivas, tanto para quem o escreveu, quanto para outros desenvolvedores. Nesse sentido, não significa que são feita apenas coisas

simples no código, mas que mesmo se for algo complexo, ainda é entendível e claro.

Manutenível

O código deve ser **fácil de modificar, corrigir e expandir** sem gerar impactos inesperados ou dificuldades excessivas. Se é possível ler e entender o código, mas não é facil de dar a manutenção, ele não é um código limpo.

Para que tenha boa Manutenibilidade os programadores precisam conseguir fazer alterações sem grandes dificuldades ou riscos. A modularização do código, padronização, testes automatizados influencia nesse aspecto.

modularização do código: Deve ser de baixo acoplamento e de alta coesão.

- Baixo acoplamento significa que os módulos do sistema devem ser independentes e evitam depender excessivamente de outros módulos
- **Alta coesão** significa que cada módulo não executa mais tarefas do que deveria.

Previsibilidade e Confiança

Para que um código seja considerado limpo, além de ser fácil de dar a manutenção, também é necessário a garantia de que as alterações feitas na manutenção não vão ser prejudiciais ao sistema, ou seja, Previsibilidade e Confiança estão relacionadas a segurança para realizar as manutenções.

Testes automatizados devem garantir que mudanças no código não causem efeitos colaterais inesperados e o código deve ser escrito de forma a minimizar surpresas e comportamentos não documentados, um código sem Previsibilidade resulta no medo de realizar alterações.

O que não é clean code

Alguns conceitos que são constantemente confundidos com Clean Code

- Não é manual: a teoria não define o clean code
- Não é estrutura de pastas

- Não é um código menor: Um código maior pode ser mais legível do que um código menor
- Não é arquitetura de software
- Não tem relação com performance

Clean Code não são regras rígidas, pode ser aplicado em qualquer estilo de diretórios, qualquer arquitetura e qualquer tecnologia. Ele não significa que o projeto ou o código vão ser pequenos e nem que vão ter um desempenho maior.

Princípios do Clean Code

O Clean Code não é um conceito que pode ser decorado, mas sim praticado e aprimorado com o tempo. Independentemente do ambiente, da equipe ou das tecnologias utilizadas, ele permite que a base do projeto cresça sem comprometer seus pilares. Por fim, os princípios do Clean Code são práticas que ajudam a alcançar seus 3 pilares.

Testes automatizados

Os testes garantem que o código funcione conforme esperado e que futuras modificações não introduzam erros, eles reduzem o medo de fazer alterações no código.

Revisão

A revisão por outros membros da equipe ajuda a identificar problemas e garantir padrões de qualidade. Ajuda a detectar erros antes que eles se tornem problemas maiores.=, além de ajudar o compartilhamento de conhecimento entre os desenvolvedores.

Refatoração

Refatorar um código significa realizar alterações que não modificam o seu comportamento, mas que o deixam melhor, como modificar os nomes de variáveis, remover duplicações e modularização de funções grandes. Nesse ponto o código é mantido limpo e organizado, além de tornado mais simples de entender.

Simplicidade

Vem do termo inglês: KISS - Keep it simple and stupid (Mantenha isso simples e idiota):
Deve-se evitar complexidades desnecessários, ou seja, só traga complexidade para o código se isso for realmente benéfico ou necessário.

Um código simples reduz a chance de bugs e Código complexo é mais difícil de entender, modificar e testar.

Iterações curtas:

Fazer pequenas mudanças e revisões frequentes para que não haja grandes blocos de códigos para serem revisados.

É difícil fazer revisões em partes grandes do código, então levar muito tempo até chegar na etapa de revisão significa um acúmulo de código além do indicado para uma boa revisão.

Seguir metodologias ágeis, como Scrum, garantem revisões frequentes do código e suas correções.