

Clean Code

- Passar por uma situação em que uma simples alteração no código quebra totalmente o sistema nos faz pensar que escrever um código totalmente novo é uma tarefa bem mais simples do que dar manutenção em código existente.
- Infelizmente, em nossas carreiras gastamos boa parte do tempo dando manutenção em código existente e, se não pensarmos direito no código que estamos escrevendo, vamos passar diversas vezes por situações semelhantes a essa.
- Lembre-se sempre que todo código que escrevemos acaba se tornando um Passivo (uma dívida) para a empresa. E quanto menos nos preocuparmos com a manutenção do código maior é o valor desse Passivo.

O que é um código com fácil manutenção?

- Quando falamos de um código com fácil manutenção estamos nos referindo à um código com Baixo Acoplamento, Alta Coesão, usando SOLID, Imutabilidade (quando fizer sentido), aplicando Design Patterns, minimizando Side Effects, maximizar o uso de Funções Puras e várias outras coisas.
- Tudo que foi citado acima pode ser resumido em ter um bom Design de Código, uma parte muito importante em ter um código limpo.
- Legibilidade conta muito para um código limpo.

Testes, testes e mais testes

- Para garantir que o código continua funcionando é necessário escrever testes.
- Testes fazem parte do jogo quando estamos desenvolvendo, e o fato de tê-los não elimina totalmente a possibilidade de termos um bug) mas minimiza bastante.
- Com os testes conseguimos garantir que pelo menos os cenários previstos estão funcionando e extrapolar esses cenários é o que torna nossos testes mais eficientes.
- Quanto mais níveis de testes (unitários, integração, aceitação, regressão e etc.) tivermos mais segurança teremos na hora de aplicar uma refatoração.
- A tarefa mais difícil na hora de se escrever um teste é saber o que devemos testar. E é justamente aí que temos que focar nossos esforços.
- Testes são uma parte importante para todo o ciclo de vida de desenvolvimento e sim, um código limpo é um código testável.

Agora sim, o que é um código limpo?

- Um código limpo é a composição de diversas características, como:

1) Legível

Um código compreensivo possibilita a identificação de pontos que precisam ser melhorados. Passamos mais tempo lendo código do que escrevendo então, quanto mais fácil for ler o código menos esforço fazemos para entendê-lo.

2) Testável

Devemos testar nossos códigos, pois isso vai nos dar segurança para podermos alterá-los. E garantir que os cenários que previmos estão de acordo com o esperado.

3) Fácil de ser mantido

Nosso código deve ser passivo de alteração tanto para adição de novas funcionalidades, quanto para aumentar a legibilidade ou manutenibilidade.

- De uma forma bem resumida um código limpo é um código testável, fácil de manter e de ler.

Exemplos de boas práticas:

- **KISS (*Keep It Simple, Stupid*, "Mantenha as coisas simples"):** Sempre que um código for escrito, ele deve ser escrito da forma mais simples possível, para manter o código mais legível. Códigos complexos demais são mais difíceis de se manter, o que é mais difícil entender o que ele faz e como ele faz.
- **DRY (*Don't Repeat Yourself*, "Não se repita"):** Todo código escrito para resolver um problema deve ser escrito apenas uma vez, a fim de evitar repetição de código. É quase uma variação do KISS, dado que a repetição de código o torna mais confuso e difícil de manter e corrigir, se necessário.