**Livro: Princípios Padrões e Práticas Ágeis Em C#**

**Autor: Robert C. Martin, Micah Martin**

**Capítulo 8: PRINCÍPIO DA RESPONSABILIDADE ÚNICA (SRP)**

O capítulo 8 sobre o Princípio da Responsabilidade Única (SRP) traz uma ideia aparentemente simples, mas que faz muito sentido na prática: **uma classe deve ter apenas um motivo para mudar**. Isso significa que, se uma classe cuida de mais de uma responsabilidade, ela acaba ficando frágil, porque uma alteração em uma parte pode quebrar outra que não tinha nada a ver.

O que chamou atenção é que, muitas vezes, a gente nem percebe que está unindo responsabilidades diferentes, porque tende a pensar nelas em grupo. Só que separar bem essas responsabilidades é o que deixa o projeto mais claro, flexível e menos propenso a problemas inesperados.

Quando uma classe concentra múltiplas responsabilidades, elas acabam ficando acopladas: ao alterar uma parte, corre-se o risco de prejudicar outra. Esse tipo de acoplamento gera projetos frágeis, difíceis de manter e de evoluir. A solução é separar as responsabilidades em **classes distintas,** tornando o sistema mais coeso, flexível e menos sujeito a erros.