UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

MODELAGEM DE PROGRAMAÇÃO ORIENTADA A OBJETOS

O Princípio da Responsabilidade Única

Aluno: Luís Carlos Ferraz

Turma: 3º Período

Professor: Giordano Ribeiro

O PRINCÍPIO DA RESPONSABILIDADE ÚNICA

"Uma classe só deve mudar por um único motivo"

Diante de um cenário, por exemplo, em que necessitamos ampliar as possibilidades de um aplicativo, onde trabalharemos numa base de código pré-existente, contamos que as "boas práticas" estejam implementadas junto a uma boa estruturação (modularização) de classes. Caso contrário, teremos um sistema frágil com um alto custo de manutenção e extensão.

Precisamos também adicionar essas novas funcionalidades com modificações mínimas no código existente, classes e métodos escritos de forma que não mais precisem ser alterados. Estamos no conceito de Aberto e Fechado, que diz: *"Entidades de software devem permanecer abertas à extensão, mas fechadas à modificação".*

Porém para atingir tal meta ou próximo dessa estruturação, precisamos seguir o Princípio da Responsabilidade Única, que nos apresenta uma forma simples e eficiente de codificação, com base que cada classe (método) tenha apenas um motivo para ser alterada (o).

Uma classe deve fazer apenas uma coisa, deve fazê-la bem e deve fazer somente ela. Se uma classe tem mais de um motivo para ser alterada, ela não segue este princípio.

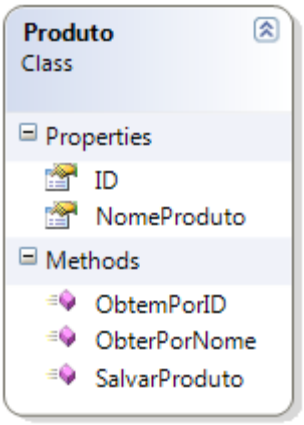
O SRP (sigla de *Single Responsability Principle*) auxilia a supressão de classes "tudo-em-um" ou "faz tudo", e também é conhecido como Princípio da Coesão.

Uma classe também deve conter um grupo coeso de propriedades e métodos, ou seja, deve haver harmonia entre as partes desta classe. Um relacionamento onde seja possível alterar cada uma destas partes de maneira independente, sem atingir outras.

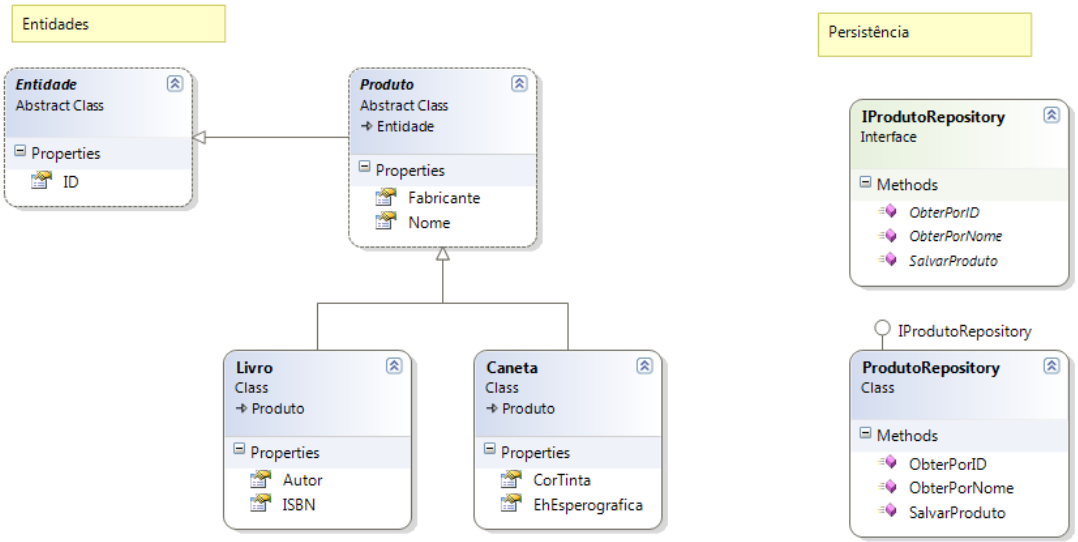
Para identificar se este princípio está sendo bem aproveitado numa certa classe devemos ser capazes de responder quais as funções que esta desempenha, e caso a resposta contenha conjunções para explicá-las, possivelmente esta classe está sendo aproveitada erroneamente. Isto não se refere à quantidade de métodos que a classe implementa e sim a coesão (relacionamento) entre estes métodos.

Sendo assim, caso a classe possua mais de uma responsabilidade, deve-se considerar uma modularização em subsistemas de classes relacionadas, cada uma com sua função específica.

Exemplo de classe com várias funções:



Exemplo de classe utilizando o SRP:



O princípio da Responsabilidade Única facilita a implementação das mudanças de requisitos, indispensável para equipes ágeis onde tais mudanças são cotidianas.