

SRP - Single Responsibility Principle

O SRP afirma que uma classe deve ter apenas uma responsabilidade e uma razão para mudar.

Benefícios:

- **Manutenção Facilitada:** Classes com uma responsabilidade específica são mais fáceis de entender e modificar.
- **Reutilização e Testes:** Classes focadas em uma única responsabilidade são mais reutilizáveis e fáceis de testar.

Exemplo de Má Prática: Uma classe que lida com lógica de negócios e também se encarrega de salvar dados no banco.

SRP - Mau Exemplo

Problema: A classe `Relatorio` tem duas responsabilidades: gerar relatório e salvá-lo. Isso viola o SRP.

```
public class Relatorio
{
    public string Conteudo { get; set; }

    public void GerarRelatorio
    {
        // Geração de relatório (lógica de negocio)
    }

    public void SalvarRelatorio
    {
        // Salvando o relatório no disco (lógica de persistência)
    }
}
```

SRP - Bom Exemplo

Divida as responsabilidades em duas classes especializadas:

- **GeradorRelatorio:** foca apenas em gerar o relatório.

- **RelatorioPersistencia**: cuida da persistencia.

```
public class GeradorRelatorio
{
    public string Gerar()
    {
        // Geração do relatório
        return conteudoRelatorio
    }
}

public class RelatorioPersistencia
{
    public void Salvar(string conteudo)
    {
        // Salvando relatorio
    }
}
```