

Introdução ao SOLID

O que esperar do curso?

- **Visão geral:**

- O que é SOLID?
- Estudo de cada princípio com aplicações práticas

- **Método de ensino:**

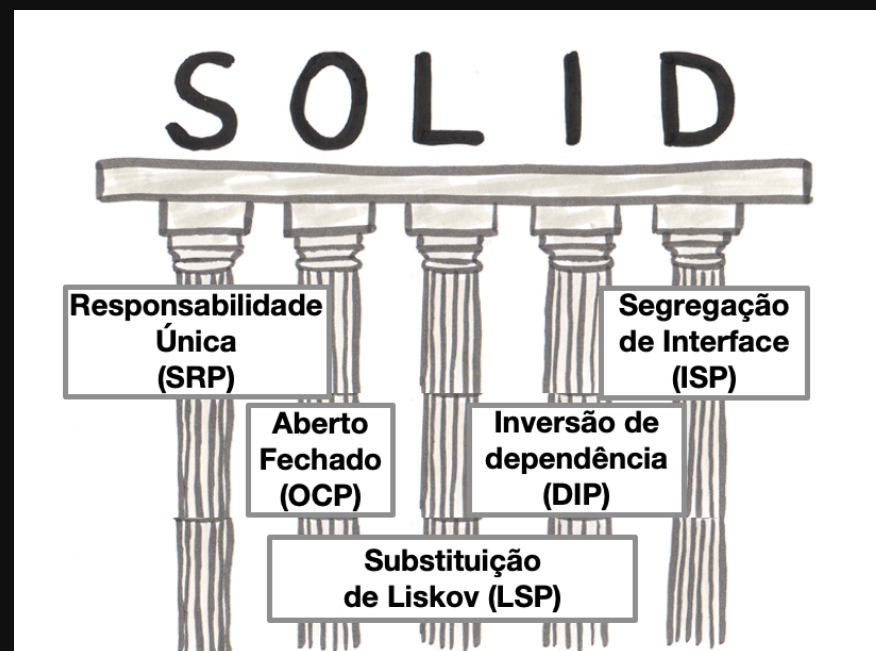
- Suporte teórico através dos slides
- Repositório git com todos os exemplos de código

O que é SOLID?

- **Definição:** é um acrônimo que representa cinco princípios fundamentais de design de software, que buscam criar sistemas compreensíveis, flexíveis e manuteníveis (sustentáveis).
- Estes princípios foram organizados e publicado por Robert C. Martin (Uncle Bob), estes princípios são de extrema relevância em programação orientada à objetos (poo)

Por que preciso estudar isto?

- Código mais **manutenível**
- Facilidade? Flexibilidade para **estender** e **evoluir** o sistema
- Maior **reusabilidade** de código
- Código mais **testável**
- Redução de **acoplamento** aferente/eferente indevido entre componentes



SRP

Single Responsibility Principle

Princípio da Responsabilidade Única

SRP

- **Conceito:** Uma classe (um componente), deve ter uma, e apenas uma, razão para mudar.

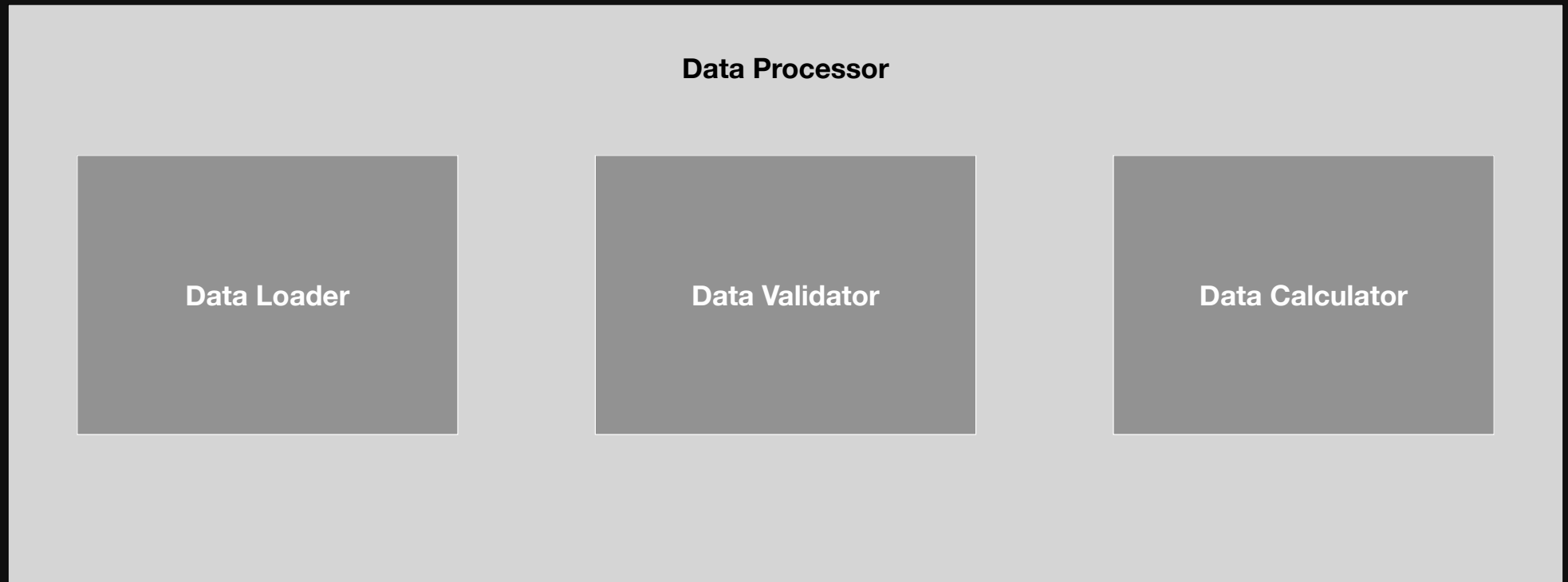
SRP – Exemplo 1

- **Exemplo teórico:** Imagine uma classe **User** que além da representação de um usuário também é responsável por salvar informações no banco de dados e enviar um e-mail.
- **Violação:** Explícita! Dificuldade de manutenção e alto grau de acoplamento

SRP – Exemplo 2

- **Exemplo teórico:** Imagine uma classe **DataProcessor** que possui 3 métodos: **Load**, **Validate** e **CalculateStatistics**. Todos estes métodos fazem parte do grupo de funcionalidades de “Processamento de dados”.
 - Load: Carrega as informações de um arquivo
 - Validate: Valida se os dados estão no formato correto e atendem as regras de negócio
 - CalculateStatistics: Calcula estatísticas com base nos dados importados
- **Violação:** Implícita! Dificuldade de manutenção e alto grau de acoplamento

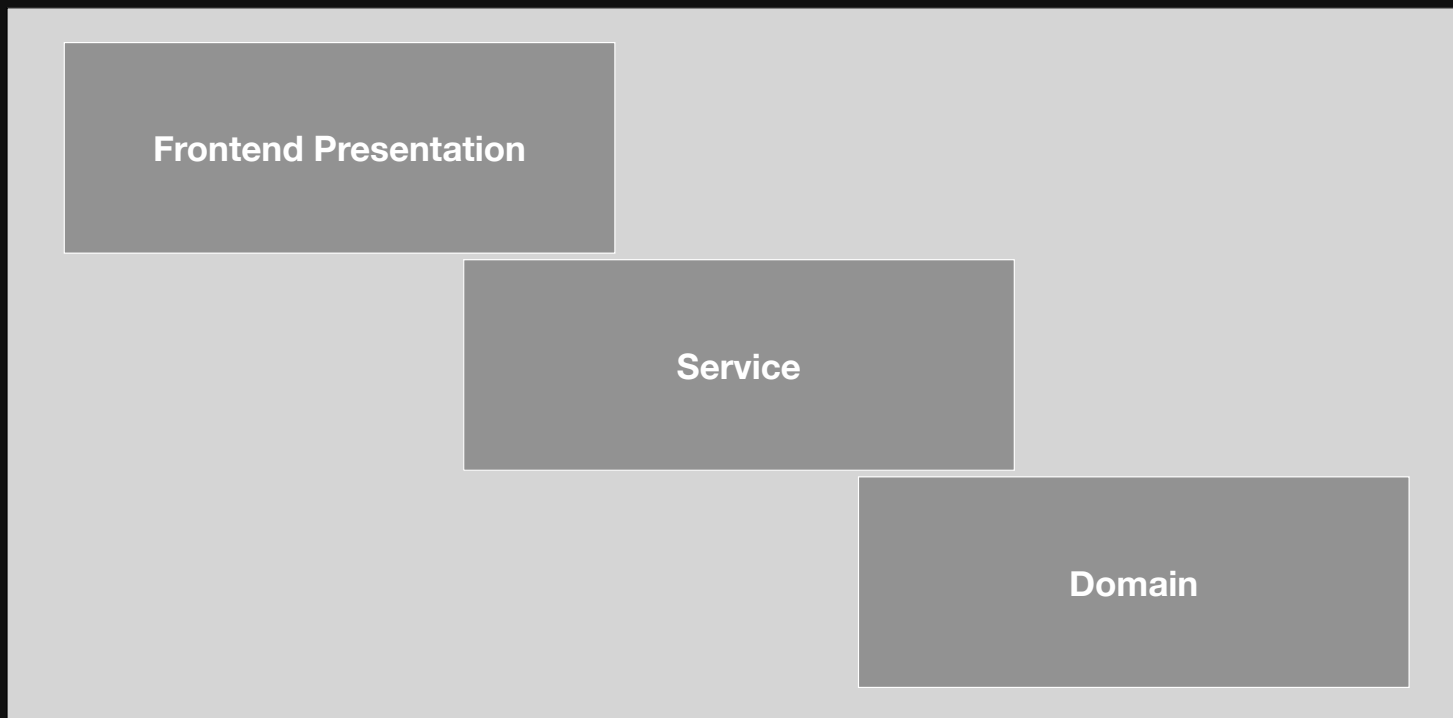
SRP – Exemplo 2



SRP – Exemplo 3

- **Exemplo teórico:** Imagine uma classe ***GetPurchaseOrderService*** que tem a função de buscar dados de um Pedido de compra.
- Contexto: A aplicação já está funcionando! Recebemos a solicitação de acrescentarmos o nome do cliente para ser exibido no frontend.
- **Violação:** Implícita! Dificuldade de manutenção e alto grau de acoplamento

SRP – Exemplo 3



OCP

Open/Closed Principle

Princípio Aberto-Fechado

OCP

- **Conceito:** As entidades de software (classes, módulos, funções) devem estar abertas para extensão, mas fechadas para modificação.

OCP – Exemplo 1

- **Exemplo teórico:** Imagine um sistema RH que calcula o salário de funcionários. Temos as seguintes classes:
 - **Employee:** Representa um funcionário
 - **EmployeeSalaryCalculator:** Faz o cálculo de salário baseado no tipo de funcionário
- **Violação:** Explícita! A cada novo tipo de remuneração, precisamos modificar o código da calculadora de salário

OCP – Exemplo 2

- **Exemplo teórico:** Imagine a geração de relatório de custos por departamento. O relatório pode ser impresso ou gerado em arquivo CSV (Comma Separated Values).
- **Violação:** Explícita! A cada novo tipo de saída, precisamos modificar o código da geração do relatório

OCP – Exemplo 3

- **Exemplo teórico:** Imagine um sistema que possui notificações e o usuário pode decidir em quais canais deseja receber a notificação.
- **Violação:** Explícita! A cada novo canal de notificação, precisamos modificar o código.