

# BANCO DO ESTADO DO RIO GRANDE DO SUL S.A. (BANRISUL)

## ÁREA 2: DESENVOLVIMENTO DE SOFTWARE

Prova Discursiva

Aplicação: 15/06/2025

### PADRÃO DE RESPOSTA DEFINITIVO

O(A) candidato(a) deverá apresentar corretamente os cinco princípios que compõem o acrônimo SOLID: **S** [*single responsibility principle* (SRP)]: cada classe deve ter apenas uma responsabilidade, ou seja, uma única razão para mudar. Isso evita que uma classe acumule muitas funções distintas, tornando-a mais simples e coesa. **O** [*open/closed principle* (OCP)]: as entidades de *software* (classes, módulos, funções) devem estar abertas para extensão, mas fechadas para modificação. Isso permite que novos comportamentos sejam adicionados sem alterar o código existente, o que reduz o risco de introdução de erros. **L** [*liskov substitution principle* (LSP)]: os objetos de uma subclasse devem ser capazes de substituir os objetos da superclasse sem alterar o comportamento correto do sistema. Isso garante integridade e previsibilidade no uso de herança. **I** [*interface segregation principle* (ISP)]: é preferível ter várias interfaces específicas e pequenas a ter uma interface única e grande. Isso evita que classes sejam obrigadas a implementar métodos que não fazem sentido para elas. **D** [*dependency inversion principle* (DIP)]: os módulos de alto nível não devem depender de módulos de baixo nível; ambos devem depender de abstrações. Isso promove um sistema desacoplado, o que facilita manutenção, testes e evolução do código.

Dos exemplos a seguir, o(a) candidato(a) deverá apresentar dois exemplos de aplicação ou de violação, como os seguintes: i) violação do SRP – uma classe `Relatorio`, que possui métodos para gerar relatórios, persistir dados em banco e enviar *e-mails*. Isso fere o SRP, pois mistura responsabilidades de geração de dados, persistência e comunicação. A correção seria dividir essa classe em três, cada uma focada em uma única responsabilidade; ii) aplicação do OCP – um sistema de cálculo de imposto que possui uma interface `ICalculoImposto` e classes, como `CalculoImpostoBrasil`, `CalculoImpostoEUA` etc. Se surge um novo país, cria-se uma nova classe que implementa essa interface, sem alterar o código existente. Isso demonstra o princípio de estar fechado para modificação, mas aberto para extensão; iii) violação do LSP – uma classe `Ave` tem um método `voar()`. Uma subclasse `Pinguim` estende `Ave`, mas não pode voar. Isso viola LSP, pois não é possível substituir `Ave` por `Pinguim` sem quebrar o funcionamento. A solução seria repensar a hierarquia, criando uma interface `IVoador` separada; iv) aplicação do ISP – em vez de ter uma interface `IMultifuncional` com métodos `imprimir()`, `digitalizar()`, `fax()`, o sistema define interfaces específicas: `IImpressora`, `IDigitalizadora`, `IFax`. Assim, uma classe `ImpressoraBasica` implementa apenas `IImpressora`; v) aplicação do DIP – uma classe `ProcessadorDePagamento` não depende diretamente de uma classe concreta, como `ServicoPayPal`, mas de uma interface `IGatewayPagamento`. Assim, é possível usar `PayPal`, `Stripe` ou outro serviço sem alterar a classe principal.

O(A) candidato(a) deverá demonstrar compreensão dos benefícios técnicos trazidos pelos princípios e deverá apresentar dois possíveis impactos, como os seguintes: i) alta manutenibilidade – o código fica mais organizado, com classes menores, mais simples e focadas, o que facilita entender e modificar; ii) melhor testabilidade – com baixo acoplamento e alto uso de abstrações, as classes podem ser facilmente testadas de forma isolada; iii) alta escalabilidade – é mais fácil adicionar novos recursos e comportamentos, pois o sistema é projetado para extensão sem alterar código existente; iv) redução de *bugs* – menos dependências ocultas e menos efeitos colaterais ao modificar uma parte do sistema; v) facilidade na refatoração e evolução – as mudanças são localizadas, sem impacto indesejado em outras partes do sistema.

#### QUESITOS AVALIADOS

##### QUESITO 2.1 – Os cinco princípios SOLID

Conceito 0 – Não atendeu ao solicitado ou o fez de forma totalmente equivocada.

Conceito 1 – Conceituou corretamente apenas um princípio.

Conceito 2 – Conceituou corretamente dois princípios.

Conceito 3 – Conceituou corretamente três princípios.

Conceito 4 – Conceituou corretamente quatro princípios.

Conceito 5 – Conceituou corretamente os cinco princípios.

##### QUESITO 2.2 – Dois exemplos de aplicação ou de violação e dois exemplos dos princípios SOLID

Conceito 0 – Não atendeu ao solicitado ou o fez de forma totalmente equivocada.

Conceito 1 – Apresentou corretamente apenas um dos exemplos de aplicação ou de violação **ou** um dos exemplos de princípios SOLID.

Conceito 2 – Apresentou corretamente apenas um dos exemplos de aplicação ou de violação **e** um dos exemplos de princípios SOLID.

Conceito 3 – Apresentou corretamente apenas um dos exemplos de aplicação ou de violação e dois exemplos de princípios SOLID; OU Apresentou corretamente dois exemplos de aplicação ou de violação e um dos exemplos de princípios SOLID.  
Conceito 4 – Apresentou corretamente dois exemplos de aplicação ou de violação e dois exemplos de princípios SOLID.

**QUESITO 2.3 – Possíveis impactos positivos de SOLID no ciclo de vida do *software***

Conceito 0 – Não atendeu ao solicitado ou o fez de forma totalmente equivocada.

Conceito 1 – Apresentou corretamente apenas um possível impacto.

Conceito 2 – Apresentou corretamente dois possíveis impactos.