

Questão 40

Uma fintech desenvolveu um sistema de processamento de pagamentos que permite a execução de múltiplas transações simultaneamente. No entanto, os desenvolvedores perceberam que, quando muitas solicitações ocorrem ao mesmo tempo, há inconsistências nos dados, como duplicação de pagamentos ou perda de registros. Para solucionar esse problema, é necessário aplicar um design pattern que controle o acesso concorrente aos recursos compartilhados, garantindo a integridade dos dados.

Qual design pattern deve ser implementado para gerenciar uma instância de controle transacional nesse sistema?

- ☐ A Factory Method.
- ☐ B Adapter.
- ☐ C Observer.
- ☒ D É a correta Singleton.
- ☐ E Strategy.

Resposta comentada	Feedback do professor
<p>O item busca aferir se o estudante compreende a necessidade de controlar o acesso concorrente a recursos compartilhados em aplicações back-end. A alternativa correta é a D, pois o padrão Singleton permite garantir que apenas uma instância de um objeto seja criada e utilizada, evitando conflitos de concorrência em ambientes de execução paralela.</p>	<p>GABARITO:</p> <p>D) Esse padrão é frequentemente usado para gerenciar o acesso a recursos compartilhados, garantindo que apenas uma instância do objeto gerenciador de transações exista, evitando duplicações ou inconsistências.</p> <p>DISTRATORES:</p> <p>A) Útil para criação de objetos de maneira flexível, mas não resolve problemas de concorrência.</p> <p>B) Utilizado para compatibilizar interfaces diferentes, sem impacto direto no controle de concorrência.</p> <p>C) Focado em notificações entre componentes, sem garantir a integridade de dados em múltiplas transações simultâneas.</p> <p>E) Define diferentes estratégias para uma funcionalidade, mas não lida diretamente com controle de concorrência.</p>