**Justificação para o Uso do Padrão Strategy no Simulador de Cenários de Exposição Radiológica**

No âmbito do projeto **Simulador de Cenários de Exposição Radiológica**, o padrão de comportamento **Strategy** foi escolhido para lidar com a necessidade de realizar diferentes análises de dados de acordo com os cenários médicos e industriais criados. Este padrão permite encapsular algoritmos relacionados, garantindo flexibilidade e modularidade na lógica do sistema.

O padrão **Strategy**, conforme descrito no livro **Gang of Four**, promove a separação entre o comportamento de um sistema e os algoritmos que podem ser aplicados dinamicamente. Define uma família de algoritmos, encapsula cada um numa classe separada, e permite que o algoritmo seja selecionado em tempo de execução. No contexto do projeto, isso permite que diferentes estratégias de análise sejam aplicadas dependendo do tipo de cenário radiológico, mantendo o código coeso e facilitando a adição de novas estratégias sem modificar o núcleo da aplicação.

**Razões para a Escolha**

1. **Flexibilidade**: A implementação do Strategy facilita a adição de novas análises (métricas específicas) sem impactar a lógica principal do sistema. Por exemplo, estratégias como **MedicalAnalysisStrategy** e **IndustrialAnalysisStrategy** foram criadas para lidar com análises específicas de cada cenário.
2. **Redução de Complexidade**: Elimina o uso de estruturas condicionais espalhadas pelo código, como múltiplos if-else ou switch-case, centralizando a lógica de seleção de algoritmos.
3. **Reutilização de Código**: As estratégias encapsulam algoritmos específicos, promovendo a reutilização e evitando duplicação.
4. **Extensibilidade**: O Strategy garante que novas estratégias podem ser integradas facilmente, como a inclusão de análises para cenários futuros sem necessidade de alterar o restante do sistema.

**Implementação no Projeto**

No sistema, a classe AnalysisContext atua como o ponto de entrada para as estratégias, delegando o cálculo das métricas às classes concretas como MedicalAnalysisStrategy e IndustrialAnalysisStrategy. O **Server REST** comunica com o contexto e, dependendo dos dados do cenário, seleciona dinamicamente a estratégia adequada. Esta abordagem mantém a lógica do servidor REST limpa e desacoplada das particularidades de cada análise.

**Alinhamento com o Gang of Four**

O livro destaca que o **Strategy** é especialmente útil em cenários onde diferentes algoritmos podem ser substituídos ou alternados dinamicamente. No projeto, isso é essencial para permitir que o sistema responda de forma eficiente a mudanças nas análises ou a requisitos novos provenientes da integração com a arquitetura Inven!RA. Além disso, a independência entre as estratégias e o contexto que as utiliza está de acordo com o princípio **Open/Closed** do design de software, permitindo que o sistema seja **aberto para extensão, mas fechado para modificações**.

**Benefícios Esperados**

* **Escalabilidade**: Adição de novos tipos de análises sem alterações substanciais na lógica existente.
* **Testabilidade**: Cada estratégia pode ser testada isoladamente, melhorando a robustez do sistema.
* **Separação de Responsabilidades**: O contexto e as estratégias são independentes, o que melhora a clareza e facilita a manutenção do código.

Dessa forma, o padrão Strategy não só atende às necessidades técnicas e arquiteturais do projeto como também alinha-se às melhores práticas descritas no capítulo do **Gang of Four**, promovendo um sistema modular, flexível e preparado para expansão.

O projeto está disponível em:

<https://github.com/lsantos1970/ActiveProvideProject.git>

**Anexos**

1. Código
2. Diagramas UML:
   * Diagrama de Componentes Strategy
   * Diagrama de Sequência Strategy

Luís Santos

1900382