**Introdução:**

No desenvolvimento do Activity Provider: Simulador de Cenários de Exposição Radiológica, identifiquei a ocorrência do antipadrão "The Blob" (God Object). Este antipadrão surge quando uma única classe concentra demasiadas responsabilidades, violando o Princípio da Responsabilidade Única (SRP) e dificultando a manutenção e escalabilidade do sistema.

**Antes da refatorização:**

A classe ScenarioFactory estava altamente acoplada e acumulava responsabilidades que deveriam ser distribuídas por componentes especializados. Esta classe era responsável por:

* Criar o ambiente e os equipamentos dos cenários médicos e industriais.
* Gerir a lógica de configuração e processamento de pedidos REST.
* Executar análises sobre os cenários.

**Após a refatorização:**

Os principais benefícios alcançados foram:

* Melhor Modularização: O código foi dividido em classes especializadas, reduzindo a complexidade da lógica central.
* Facilidade de Manutenção: Novos cenários ou análises podem ser adicionados sem modificar a estrutura existente.
* Menor Acoplamento: O servidor REST não depende diretamente da implementação das fábricas ou das estratégias de análise.
* Melhoria no Teste e Reutilização: A separação de responsabilidades permite a implementação de testes unitários específicos para cada componente.

**Conclusão:**

A identificação e refatorização do antipadrão The Blob no Activity Provider demonstrou a importância de boas práticas de design para a evolução sustentável de um sistema.

Com estas melhorias, em conjunto com a adoção dos padrões Abstract Factory, Adapter e Strategy , o sistema está melhor preparado para futuras expansões e integração com novas funcionalidades dentro da plataforma Inven!RA.

O projeto está disponível em:

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

Anexos

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