

Relatório – Detecção e Refatorização de Antipadrões

Projeto: Activity Provider

Autor: Bruno Manuel

Data: 08/02/2026

1. Objetivo

Identificar a ocorrência de um antipadrão de desenvolvimento no projeto Activity Provider e aplicar uma refatorização adequada, mantendo o comportamento funcional do sistema e os padrões de desenho já presentes.

2. Enquadramento do Projeto

O Activity Provider recebe pedidos do cliente externo (Inven!RA), adapta a estrutura do pedido por meio de um Adapter e processa o fluxo através de uma cadeia de handlers (Chain of Responsibility). O conteúdo de analytics é construído com apoio de uma fábrica (Factory Method) e devolvido ao cliente em formato JSON.

3. Antipadrão Detetado

Antipadrão identificado: Cut-and-Paste Programming.

Na implementação inicial, existia tendência para agregação e repetição de blocos semelhantes (handlers e montagem da cadeia), o que favorece alterações duplicadas, divergências entre classes e aumento de acoplamento do ponto de entrada ao detalhe interno.

4. Justificação (por que é um antipadrão)

Cut-and-Paste Programming ocorre quando partes semelhantes do sistema são implementadas por duplicação. Isso aumenta o custo de manutenção (cada alteração precisa ser repetida), cria risco de inconsistências e dificulta a evolução do projeto (por exemplo, inserir um novo handler ou alterar a ordem do pipeline).

5. Refatorização Aplicada

Medidas executadas:

- Separação dos handlers em ficheiros individuais (uma classe por ficheiro), reduzindo agregação e favorecendo modularidade.
- Introdução de uma PipelineFactory para construir a cadeia de handlers num único ponto.
- Simplificação do ponto de entrada (index.php), que passa a apenas criar o contexto, pedir o pipeline e executar o handle.
- Organização do código por responsabilidades (adapters, handlers, factories e repositories).

6. Resultado e Benefícios

Após a refatorização, o projeto apresenta menor acoplamento no ponto de entrada, menor risco de divergência entre handlers e maior facilidade de manutenção. A montagem do pipeline ficou centralizada, tornando simples ajustar a sequência ou incluir novos passos de processamento.

7. Diagramas

7.1 Diagrama de Componentes (após refatorização)

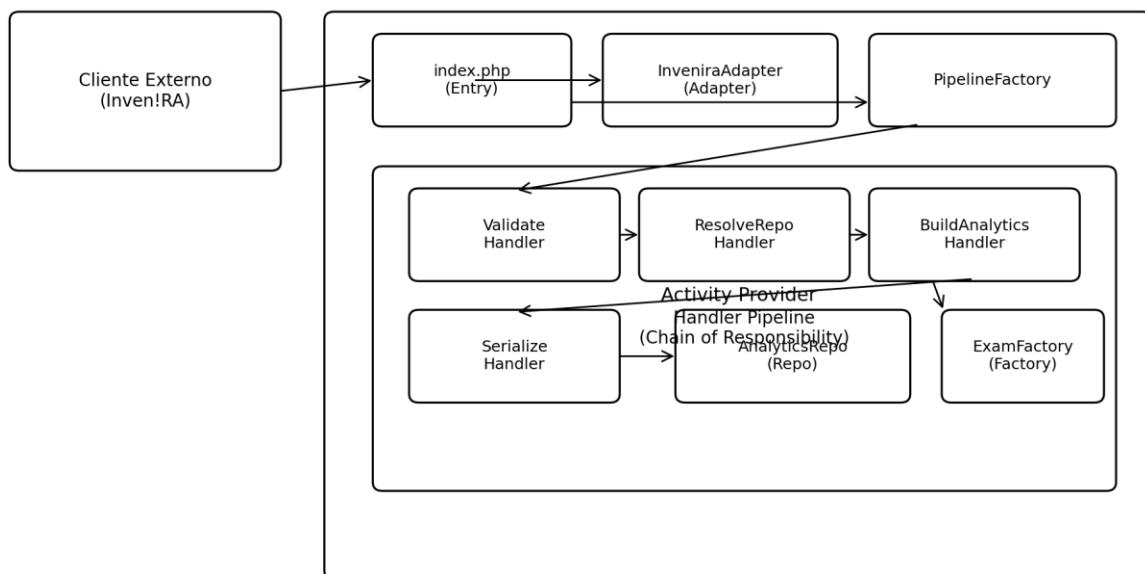


Figura 1 — Diagrama de Componentes (após refatorização)

7.2 Diagrama de Sequência (após refatorização)

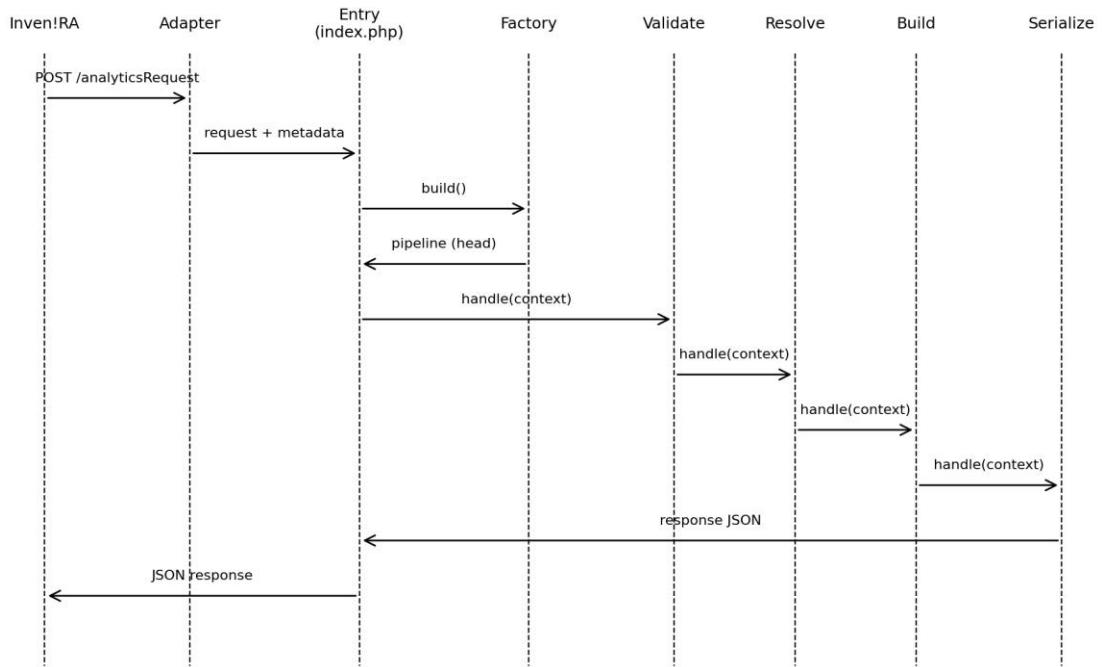


Figura 2 — Diagrama de Sequência (após refatorização)

8. Conclusão

O antipadrão Cut-and-Paste Programming foi mitigado com modularização e centralização da montagem do pipeline. A solução final mantém o funcionamento do Activity Provider e melhora a clareza estrutural, favorecendo evolução e manutenção.