

Mestrado em Engenharia Informática e Tecnologia Web

Arquitetura e Padrões de Software
(22304)
Ano letivo 2025/2026

Aplicação do Padrão de Estrutura *Facade* no ProPlan

Síntese

ProPlan é um módulo *Activity Provider* que apresenta ao aluno uma série de desafios de gestão de projeto simulados, baseados em cenários realistas.

Aplicação do padrão estrutural *Facade* (T5)

1. Enquadramento e objetivo

No âmbito da T5 – *Padrões Estruturais*, procedeu-se à aplicação do padrão Facade no *Activity Provider* ProPlan, integrado na arquitetura Inven!RA.

O objetivo principal foi reduzir o acoplamento entre a camada de endpoints HTTP (Flask) e o subsistema interno responsável pela validação de pedidos, seleção de repositórios e preparação das respostas, garantindo simultaneamente coerência entre o contrato de *analytics* e os valores efetivamente devolvidos.

A opção pelo padrão Facade resulta da identificação de um problema arquitetural concreto: a dispersão da lógica de orquestração pelos endpoints, que conheciam diretamente fábricas, repositórios e estruturas de dados.

2. Alterações introduzidas no projeto

2.1. Alterações ao código existente

Foram alterados apenas os seguintes endpoints, de forma incremental e testada em produção (Render):

- **POST /analytics-proplan**

Passou a delegar no ProPlanServiceFacade, ficando o endpoint responsável apenas por:

- parsing do pedido;
- devolução do código HTTP apropriado.

- **GET /analytics-list-proplan**

Deixou de aceder diretamente à constante ANALYTICS_SCHEMA, passando o contrato de *analytics* a ser fornecido pelo Facade, reforçando a governação centralizada do subsistema.

O padrão de criação Factory Method, previamente implementado, foi mantido sem alterações, sendo agora orquestrado pelo Facade.

2.2. Novos ficheiros introduzidos

Foram adicionados os seguintes ficheiros ao repositório:

- **services/proplan_facade.py**

Implementa o componente ProPlanServiceFacade, responsável por:

- validação dos pedidos;

- coordenação do *Factory Method*;
- obtenção de dados via repositórios;
- normalização das respostas.

- **serializers/analytics_serializer.py**

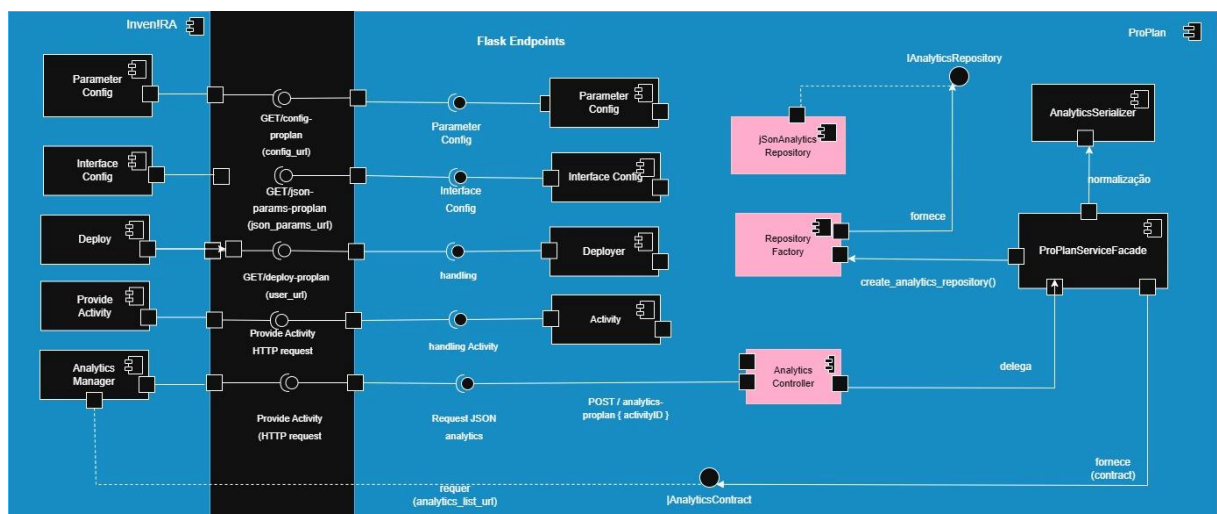
Centraliza a normalização/serialização das respostas de *analytics*. Nesta fase apresenta comportamento *pass-through*, mas prepara o sistema para evolução futura.

- **exceptions.py**

Define exceções de validação (ex.: *InvalidRequestError*), mapeadas para respostas HTTP 400, reforçando a clareza do fluxo de erro.

3. Diagrama de Componentes (versão final)

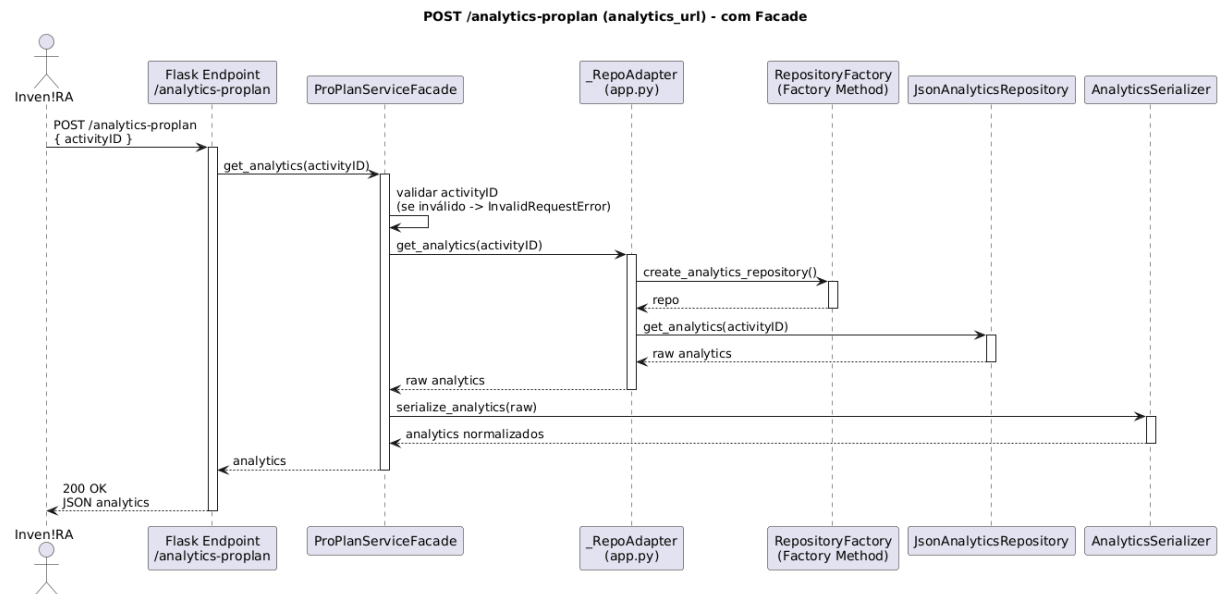
O diagrama seguinte reflete a estrutura final do *Activity Provider*, após a introdução do padrão Facade.



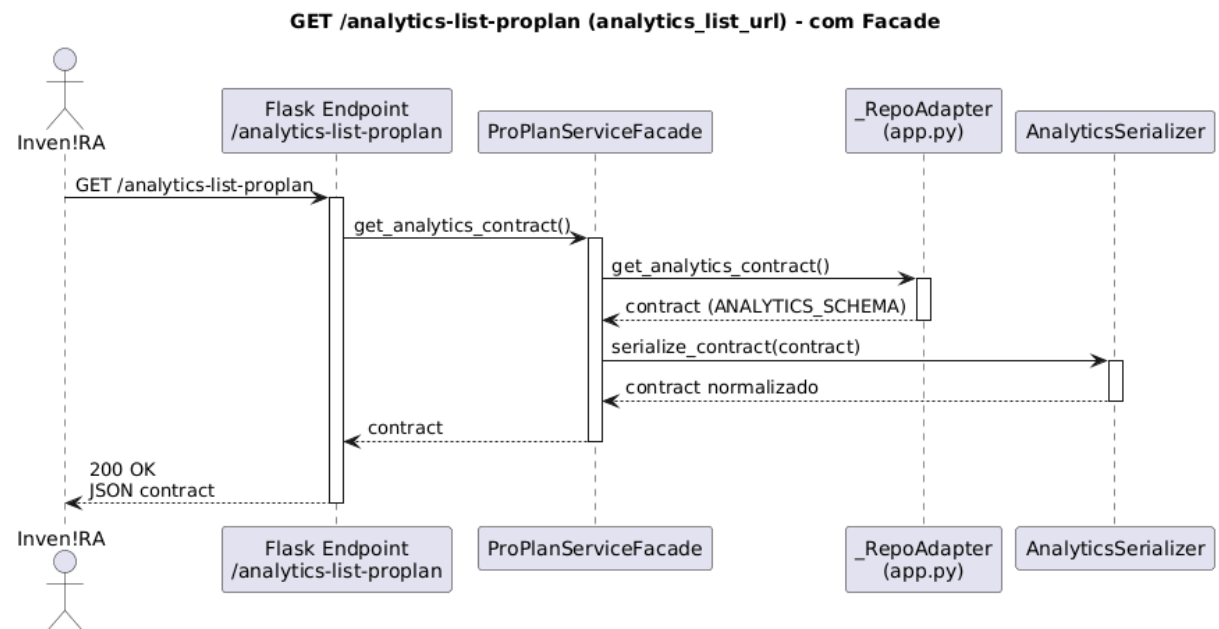
O Facade encontra-se virado para dentro, não sendo visível à Inven!RA, funcionando como ponto único de orquestração interna.

4. Diagramas de Sequência

4.1. Sequência — POST /analytics-proplan



4.2. Sequência — GET /analytics-list-proplan



5. Considerações finais

A introdução do padrão Facade permitiu:

- reduzir o acoplamento entre endpoints e subsistema interno;
- concentrar regras e compromissos num único ponto arquitetural;
- reforçar a coerência entre o contrato anunciado e os dados devolvidos;
- preparar o sistema para evolução futura sem impacto nos endpoints.

O padrão foi aplicado de forma incremental, validado em produção e articulado com o Factory Method previamente existente, resultando numa solução arquitetural clara, justificada e alinhada com os princípios do *Design Patterns*.

Referências bibliográficas (ABNT)

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. Design patterns: elements of reusable object-oriented software. Reading: Addison-Wesley, 1995.

MORGADO, L.; CASSOLA, F. Activity Providers na Inven!RA. Lisboa: Universidade Aberta, 2022.

PLATAFORMABERTA. Pedidos da Inven!RA e formatos de dados. Universidade Aberta, 2025.

SOUSA, André. *ProPlan – Activity Provider*. GitHub, 2025. Disponível em: <https://github.com/AndreMacielSousa/proplan-activity-provider>. Acesso em: 27 dez. 2025.