UNIVERSIDADE ABERTA www.uab.pt

UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

Mestrado em Engenharia Informática e Tecnologia Web

Arquitetura e Padrões de Software

(22304)

Ano letivo 2025/2026

Aplicação do Padrão de Comportamento Observer 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.

André Sousa

1300012@estudante.uab.pt

Repositório Projecto: AndreMacielSousa/proplan-activity-provider: proplan-activity-provider

APLICAÇÃO DO PADRÃO DE COMPORTAMENTO OBSERVER NO PROPLAN

1. Objetivo das alterações

As alterações introduzidas nesta fase tiveram como objetivo **aplicar explicitamente um padrão comportamental** ao projeto *ProPlan Activity Provider*, dando continuidade ao trabalho desenvolvido nas fases anteriores (Factory Method e Facade), **sem alterar o comportamento observável dos endpoints REST** definidos pela especificação Inven!RA.

Pretendeu-se, em particular:

- tornar explícita a natureza **reativa e orientada a eventos** do Activity Provider;
- desacoplar responsabilidades transversais do fluxo principal de orquestração;
- garantir que o padrão aplicado é **intencional, verificável no código e refletido nos diagramas**.

2. Enquadramento arquitetural

No estado anterior do projeto, o componente ProPlanServiceFacade já concentrava a orquestração interna dos pedidos, mantendo os endpoints Flask reduzidos a validação de parâmetros e tratamento HTTP.

Este componente revelou-se o ponto arquitetural adequado para a introdução do padrão **Observer**, assumindo o papel de **Subject**, uma vez que:

- centraliza operações relevantes do ciclo de vida da atividade (deploy e pedidos de analytics);
- não deve conhecer nem depender de funcionalidades transversais, como registos, contadores ou rastos qualitativos.

3. Alterações introduzidas no código

3.1. Infraestrutura do Observer

Foi introduzida uma infraestrutura mínima e explícita do padrão Observer através do ficheiro:

- services/events.py

Este ficheiro define:

- o DomainEvent, representando eventos de domínio relevantes;
- o contrato Observer;

- o EventPublisher, responsável pela gestão de observadores e notificação de eventos.

O componente ProPlanServiceFacade passou a herdar de EventPublisher, assumindo formalmente o papel de **ConcreteSubject**.

3.2. Emissão de eventos no Facade

Foram introduzidos eventos explícitos de ciclo de vida no ProPlanServiceFacade:

- ActivityDeployed — emitido após o deploy de uma instância;
- AnalyticsRequested — emitido sempre que é solicitado o serviço analytics_url.

Estes eventos são emitidos sem alterar o valor devolvido pelos métodos, garantindo que o comportamento externo do serviço permanece inalterado.

3.3. Observadores concretos

Foram implementados três *ConcreteObservers* no ficheiro:

- services/observers.py

Nomeadamente:

- DeployRegistryObserver — regista metadados mínimos do deploy por activityID;
- AnalyticsRequestCounterObserver — contabiliza pedidos de analytics por instância;
- DecisionLogObserver — mantém um rasto textual (mock) de eventos relevantes.

Estas implementações são deliberadamente leves e em memória, sendo suficientes para demonstrar o padrão no contexto atual do projeto.

3.4. Ligação dos observadores

Os observadores são instanciados e registados (attach) no arranque da aplicação, no ficheiro app.py.

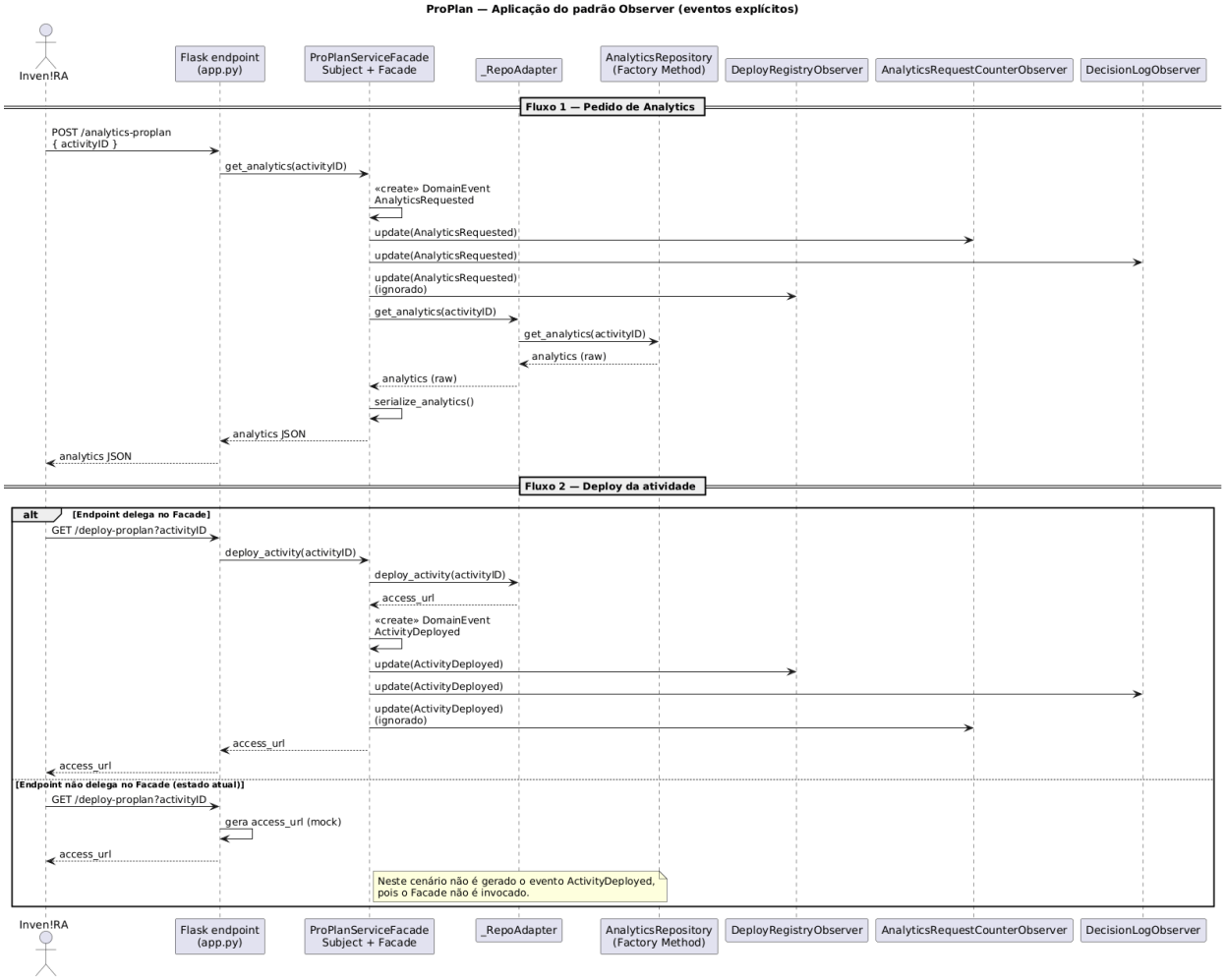
Esta ligação garante que:

- o padrão está ativo em runtime;
- os eventos emitidos pelo Facade são efetivamente observados;
- não existe acoplamento direto entre o Facade e os observadores.

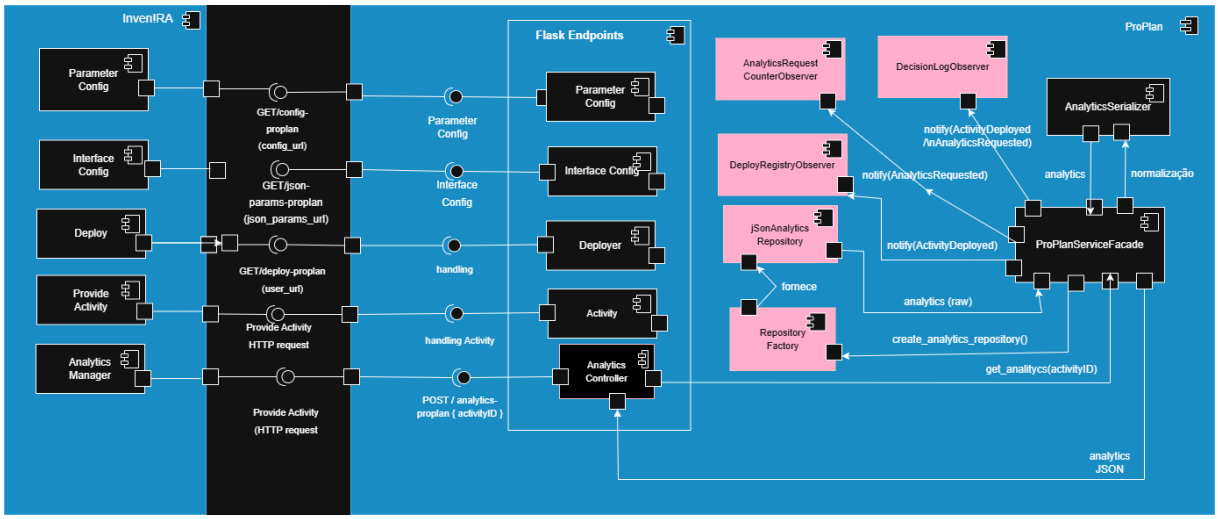
4. Diagramas apresentados

Para acompanhar o código, foram elaborados dois diagramas UML:

- **Diagrama de Sequência** — ilustra a aplicação do padrão Observer nos fluxos de analytics-proplan e, de forma condicional, no fluxo de deploy, evidenciando a criação e propagação de eventos;



- **Diagrama de Componentes** — representa exclusivamente os componentes relevantes para o padrão Observer, destacando o ProPlanServiceFacade como Subject e os observadores concretos registados no sistema.



Em ambos os casos, foram deliberadamente excluídos elementos não relevantes para o padrão, de forma a manter clareza e foco arquitetural.

5. Impacto no comportamento do sistema

As alterações introduzidas **não modificam**:

- os endpoints disponíveis;
- os contratos REST;
- os formatos das respostas;
- o comportamento observado pelos consumidores do serviço.

O impacto é exclusivamente **interno e arquitetural**, preparando o sistema para evolução futura sem comprometer a compatibilidade com a especificação Inven!RA.

6. Considerações finais

A aplicação do padrão Observer reforça a coerência do desenho do ProPlan enquanto Activity Provider orientado a eventos, permitindo a introdução incremental de

funcionalidades transversais sem degradação da legibilidade ou aumento do acoplamento.

O padrão foi aplicado de forma explícita, limitada ao contexto real do projeto e suportada por código e diagramas coerentes, cumprindo os objetivos definidos para esta fase da unidade curricular.

Referências

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.