

Resenha: Arquitetura Hexagonal (Ports & Adapters)

Autor: Alistair Cockburn (2005)

Fonte: <https://alistair.cockburn.us/hexagonal-architecture/>

A Arquitetura Hexagonal, também conhecida como padrão Ports and Adapters, foi proposta por Alistair Cockburn em 2005 com o objetivo de separar o núcleo de negócios (core) das dependências externas, como banco de dados, interfaces de usuário ou sistemas de terceiros. Essa abordagem propõe que a aplicação seja independente da infraestrutura, facilitando testes, manutenção e evolução do sistema.

Cockburn argumenta que a maioria dos sistemas acaba misturando regras de negócio com detalhes técnicos, o que causa acoplamento e dificulta a testabilidade. Para resolver isso, ele sugere o uso de 'portas' (ports) e 'adaptadores' (adapters). As portas representam interfaces de comunicação entre o sistema e o mundo externo, enquanto os adaptadores implementam as conexões com as tecnologias específicas.

Visualmente, o modelo é representado por um hexágono, onde cada lado simboliza uma interface de entrada ou saída. Essa forma destaca que o sistema pode ser acessado por múltiplos tipos de agentes, como usuários, scripts, testes automatizados ou outros sistemas. O número de lados não é fixo — o hexágono é apenas uma metáfora visual para expressar a ideia de múltiplas conexões.

O padrão define dois tipos principais de adaptadores: os 'primários', responsáveis por acionar a aplicação (por exemplo, uma API REST ou interface web), e os 'secundários', controlados pela aplicação (como banco de dados, filas ou APIs externas). Essa separação garante que o domínio permaneça isolado, podendo ser testado com facilidade utilizando mocks ou adaptadores de teste.

Os principais benefícios do modelo são: testabilidade elevada, facilidade na substituição de tecnologias, maior clareza de responsabilidades e manutenibilidade aprimorada. Contudo, o autor reconhece que para sistemas simples a aplicação do padrão pode ser considerada exagerada, pois adiciona certa complexidade estrutural.

A Arquitetura Hexagonal influenciou diretamente conceitos modernos como a Arquitetura Limpa e o uso de camadas em frameworks modernos como Laravel, Spring Boot e Flask. O foco em isolar o domínio e permitir testes independentes reflete práticas essenciais da engenharia de software moderna.

Em síntese, a proposta de Cockburn é uma contribuição fundamental para o desenvolvimento de sistemas desacoplados e sustentáveis. Ela incentiva os desenvolvedores a pensar em termos de interfaces e dependências invertidas, criando

aplicações mais flexíveis e resistentes a mudanças tecnológicas.

Conclusão:

O artigo de Alistair Cockburn oferece uma base sólida sobre o design de software modular e sustentável. A ideia de que a lógica de negócio deve ser independente das tecnologias externas é um princípio que continua extremamente relevante, especialmente para quem busca escrever código limpo, testável e preparado para o futuro.