

Resenha do artigo “The Hexagonal (Ports & Adapters) Architecture” – Alistair Cockburn (2005)

O artigo apresenta a **Arquitetura Hexagonal**, também conhecida como **Ports and Adapters**, como uma alternativa às arquiteturas tradicionais em camadas, com o objetivo de **desacoplar a lógica de negócio das interfaces externas**, como bancos de dados, interfaces gráficas ou serviços de terceiros. Alistair Cockburn critica a recorrente contaminação da regra de negócio por detalhes de implementação externos, o que dificulta testes automatizados, manutenibilidade e evolução do sistema.

A proposta central é tratar a aplicação como um **núcleo independente** (o "lado de dentro"), que se comunica com o mundo externo por meio de **portas (ports)**. Cada porta representa um contrato ou protocolo, e os **adapters** são peças plugáveis responsáveis por traduzir as interações entre o núcleo da aplicação e diferentes tecnologias. Assim, uma mesma porta pode ter vários adaptadores: por exemplo, uma porta de entrada pode aceitar uma interface gráfica, uma API HTTP ou um script de testes automatizados.

Cockburn argumenta que essa separação favorece **testabilidade**, permitindo rodar a aplicação com bancos de dados falsos (mocks) ou interfaces simuladas, sem dependência de infraestrutura. Além disso, facilita a substituição de tecnologias ao longo do tempo, já que a aplicação passa a depender apenas de contratos, e não de implementações específicas.

O autor reforça a simetria da arquitetura ao explicar que tanto entradas (usuário, API, testes) quanto saídas (bancos de dados, mensagens, integração com outros sistemas) são tratadas da mesma forma: ambas são adaptadores conectados a portas, eliminando a visão engessada de camadas verticais ou horizontais. A metáfora do hexágono não tem importância geométrica, mas visual – serve apenas para representar múltiplos pontos de conexão possíveis.

Ao longo do texto, Cockburn demonstra a aplicabilidade do modelo com exemplos práticos, mostrando como adicionar gradualmente GUI, testes automatizados e banco real, sempre mantendo a aplicação central isolada. Ele também relaciona a arquitetura com padrões conhecidos como **Adapter**, **MVC**, **Dependency Injection** e o uso de **Mocks**, evidenciando que a proposta dialoga com outras práticas de desenvolvimento orientado a testes e modularidade.

Em síntese, o artigo defende que **a verdadeira independência da lógica de negócio só é atingida quando a aplicação pode ser executada sem interface gráfica, banco de dados ou integrações diretas**, funcionando completamente isolada através de portas bem definidas. Com isso, a Arquitetura Hexagonal se destaca como um modelo que promove flexibilidade, facilidade de manutenção e alta testabilidade, alinhando-se aos princípios da engenharia de software moderna e desenvolvimento sustentável de sistemas.