

O artigo “The Hexagonal (Ports & Adapters) Architecture”, de Alistair Cockburn (2005), apresenta uma proposta de arquitetura de software que busca isolar a aplicação de dependências externas, como interfaces gráficas e bancos de dados. O autor propõe uma estrutura que permita ao sistema ser testado, executado e evoluído independentemente de tecnologias específicas, garantindo maior flexibilidade e manutenibilidade.

Cockburn inicia o texto destacando um problema recorrente: a mistura entre regras de negócio e código de interface. Essa prática torna o sistema frágil, difícil de testar automaticamente e dependente de detalhes visuais ou de infraestrutura. Para resolver isso, ele introduz o conceito de “ports and adapters” (portas e adaptadores), que mais tarde ficou conhecido como arquitetura hexagonal. A ideia é que a aplicação possua um núcleo independente — o “interior” — que se comunica com o “exterior” por meio de portas. Cada porta define um contrato (interface) que pode ser implementado por diferentes adaptadores, permitindo trocar tecnologias sem alterar a lógica central.

O artigo explica que as portas podem ser divididas em primárias (as que recebem comandos, como uma interface de usuário ou uma API) e secundárias (as que o sistema utiliza para obter dados, como um repositório ou serviço externo). Essa separação cria simetria e independência entre o domínio e o ambiente, tornando o sistema testável com ferramentas automatizadas e “mock databases” sem precisar de infraestrutura real.

Ao longo do texto, Cockburn apresenta exemplos práticos, como um sistema simples de cálculo de descontos, ilustrando a evolução da aplicação desde o uso de testes automatizados até a integração com bancos de dados reais. Ele também relaciona a arquitetura hexagonal com outros padrões e princípios, como Adapter, Model-View-Controller (MVC), Dependency Injection e Mock Objects, mostrando que a proposta não substitui esses conceitos, mas os complementa em um nível arquitetural.

Na parte final, o autor discute casos de uso e variações da abordagem, ressaltando sua aplicação em sistemas complexos e distribuídos. Ele destaca ainda que o número de portas pode variar conforme o contexto, e que o principal valor do padrão está na isolação do núcleo da aplicação — permitindo que ela funcione mesmo sem interface gráfica, banco de dados ou serviços externos.

Em síntese, o artigo de Cockburn representa um marco conceitual na engenharia de software ao propor uma arquitetura que separa claramente as preocupações de negócio das de infraestrutura. A arquitetura hexagonal tornou-se uma referência fundamental para o desenvolvimento de sistemas modulares, testáveis e adaptáveis, influenciando diretamente outras abordagens modernas, como Clean Architecture e Onion Architecture.