

Hexagonal Architecture

A Arquitetura Hexagonal, também conhecida como "Portas e Adaptadores" (Ports and Adapters), é um padrão de design de software proposto por Alistair Cockburn que visa criar uma separação rigorosa entre a lógica de negócios central de uma aplicação e os componentes externos dos quais ela depende. O objetivo é isolar o núcleo da aplicação de elementos como interfaces de usuário (UI), bancos de dados, APIs de terceiros ou scripts de teste, permitindo que esse núcleo seja desenvolvido e testado de forma totalmente independente.

Historicamente, um dos maiores problemas no desenvolvimento de software é a infiltração da lógica de negócios em outras camadas, especialmente na interface do usuário e no código de acesso a dados. Isso torna as aplicações frágeis, pois qualquer mudança visual na UI ou alteração no banco de dados pode quebrar a lógica central. Além disso, essa dependência impede testes automatizados eficientes, já que a lógica de negócios não pode ser testada sem que a UI ou o banco de dados estejam presentes e funcionando.

A solução da Arquitetura Hexagonal é inverter o pensamento tradicional de camadas (UI \rightarrow Lógica \rightarrow Dados) e focar na assimetria entre o "interior" (o domínio de negócios) e o "exterior" (a infraestrutura e tecnologia). O núcleo da aplicação, o "hexágono", contém apenas a lógica de negócios pura, sem qualquer conhecimento sobre como será usado ou quais tecnologias o cercam.

Essa comunicação entre o interior e o exterior é gerenciada por dois componentes principais. As "Portas" (Ports) são as APIs da aplicação, definidas como interfaces que especificam a *intenção* de uma interação (como [CalcularDesconto](#) ou [SalvarCliente](#)), sem ditar a tecnologia. Os "Adaptadores" (Adapters) são as pontes tecnológicas que ficam no exterior; eles traduzem a linguagem de uma tecnologia específica (como uma requisição HTTP de uma API REST ou uma consulta SQL) para o formato neutro definido pela Porta.

Essa abordagem se aplica simetricamente a todos os lados da aplicação. Existem os "Atores Primários" (ou Driving Adapters), que *iniciam* uma ação na aplicação, como uma interface gráfica, um script de teste ou outra aplicação. Todos eles se conectam à mesma Porta de entrada. Da mesma forma, existem os "Atores Secundários" (ou Driven Adapters), que *são chamados* pela aplicação quando ela precisa de algo, como um banco de dados para buscar dados ou um serviço de e-mail para enviar uma notificação. A aplicação central apenas chama uma Porta de saída (uma interface), e diferentes adaptadores—seja um adaptador para um banco SQL real ou um "mock" de teste em memória—podem ser "plugados" para implementar essa interface.

O nome "Hexagonal" não tem relação com o número seis; é uma metáfora visual para quebrar o pensamento linear de "cima para baixo" das arquiteturas em camadas, permitindo visualizar a aplicação com múltiplos "lados" ou pontos de conexão, todos tratados de forma equivalente.

Os benefícios dessa arquitetura são imensos. O principal é a **testabilidade total**, pois a lógica de negócios pode ser testada em completo isolamento, usando adaptadores de teste e mocks, resultando em testes rápidos e robustos. Isso leva ao **agnosticismo tecnológico**, onde o núcleo não sabe se está sendo usado por uma aplicação web ou um script, ou se seus dados vêm de um banco SQL ou um arquivo. Consequentemente, a aplicação ganha enorme **flexibilidade e manutenção**, permitindo trocar um banco de dados ou uma interface de usuário inteira apenas escrevendo um novo adaptador, sem jamais tocar na lógica de negócios central.