

Arquitetura Hexagonal (Ports & Adapters) de Alistair Cockburn

A Arquitetura Hexagonal, também conhecida como "Ports & Adapters" (Portas e Adaptadores), é um padrão de arquitetura de software criado por Alistair Cockburn com o objetivo de criar sistemas com baixo acoplamento e alta coesão, isolando a lógica de negócio das dependências externas, como interfaces de usuário, bancos de dados, e outros sistemas.

O Problema que a Arquitetura Hexagonal Resolve

Em arquiteturas tradicionais em camadas, como o MVC (Model-View-Controller), é comum que a lógica de negócio fique fortemente acoplada à interface do usuário (UI) ou ao banco de dados. Isso torna o sistema difícil de testar, manter e evoluir. Por exemplo, para testar a lógica de negócio, muitas vezes é necessário ter um banco de dados em execução ou simular requisições HTTP, o que complica o processo.

A Solução: Portas e Adaptadores

A ideia central da Arquitetura Hexagonal é inverter essa dependência. A lógica de negócio (o "hexágono") não depende de nada externo. Em vez disso, ela define "portas" (interfaces) que especificam como ela quer interagir com o mundo exterior.

- **O Hexágono:** Representa o núcleo da aplicação, contendo a lógica de negócio e as regras de domínio. Ele é completamente independente de qualquer tecnologia externa.
- **Portas (Ports):** São as interfaces que definem um contrato de comunicação entre o núcleo da aplicação e o mundo exterior. Existem dois tipos de portas:
 - **Portas de Entrada (Driving Ports):** Definem como o mundo exterior pode interagir com a aplicação (por exemplo, uma interface para um serviço de usuário).
 - **Portas de Saída (Driven Ports):** Definem como a aplicação interage com o mundo exterior (por exemplo, uma interface para um repositório de banco de dados).
- **Adaptadores (Adapters):** São as implementações concretas das portas. Eles traduzem a comunicação entre o mundo exterior e o núcleo da aplicação.
 - **Adaptadores de Entrada (Driving Adapters):** Conectam a aplicação a diferentes tipos de clientes. Por exemplo, um controlador de API REST, uma interface de linha de comando (CLI) ou um consumidor de mensagens.
 - **Adaptadores de Saída (Driven Adapters):** Conectam a aplicação a diferentes tecnologias de infraestrutura. Por exemplo, um adaptador para um banco de dados SQL, um cliente de e-mail ou um cliente de API externa.

Por que "Hexagonal"?

O nome "Hexagonal" não tem a ver com o número seis. Cockburn escolheu essa forma para ter espaço suficiente no diagrama para representar as várias portas e adaptadores, evitando a visualização tradicional de "cima para baixo" ou "esquerda para a direita" das

arquiteturas em camadas e destacando a simetria entre as diferentes formas de interagir com a aplicação.

Benefícios da Arquitetura Hexagonal

- **Testabilidade:** A lógica de negócio pode ser testada de forma isolada, sem a necessidade de um banco de dados ou interface de usuário.
- **Independência de Frameworks:** A aplicação não fica amarrada a um framework específico, facilitando a troca de tecnologias.
- **Flexibilidade:** Permite que a aplicação seja controlada por diferentes tipos de clientes (web, mobile, CLI, testes automatizados) e se conecte a diferentes tecnologias de infraestrutura.
- **Separação de Responsabilidades:** Mantém uma separação clara entre a lógica de negócio e as preocupações de infraestrutura.

Conclusão

A Arquitetura Hexagonal é uma abordagem poderosa para o design de software que promove a criação de sistemas robustos, flexíveis e fáceis de manter. Ao isolar o núcleo da aplicação, ela permite que o software evolua de forma mais segura e sustentável, adaptando-se às mudanças tecnológicas sem impactar a lógica de negócio. É um padrão fundamental para quem busca construir aplicações de alta qualidade e longa duração.