

Desacoplando o Núcleo: Uma Análise sobre "Hexagonal Architecture"

No desenvolvimento de software, uma das dores de cabeça mais comuns é o emaranhado que se forma entre a lógica de negócio da nossa aplicação e os detalhes de infraestrutura. É a regra de negócio que sabe que precisa de um banco de dados SQL, a lógica de domínio que depende de uma API REST específica, e o núcleo do sistema que fica poluído com código de interface de usuário. Essa mistura, embora muitas vezes seja o caminho mais rápido no início, inevitavelmente nos leva a um beco sem saída, onde cada mudança se torna um pesadelo e os testes se tornam frágeis e complexos. O artigo "Hexagonal Architecture", de Alistair Cockburn, ataca a raiz desse problema com uma proposta elegante para proteger o que realmente importa: a nossa aplicação.

Cockburn nos convida a reimaginar a estrutura de um sistema. Em vez de pensar em camadas empilhadas umas sobre as outras (UI em cima, negócio no meio, dados embaixo), ele nos propõe a imagem de um hexágono. No centro desse hexágono vive o coração do nosso sistema: a aplicação, com suas regras de negócio e lógica de domínio, completamente pura e ignorante sobre o mundo exterior. Ela não sabe se será usada por um usuário através de uma interface web, por um sistema de testes automatizados ou por um script de linha de comando. Ela também não se importa se seus dados serão persistidos em um banco de dados Oracle, em um MongoDB ou em arquivos de texto.

A grande sacada da Arquitetura Hexagonal, também conhecida como "Ports and Adapters" (Portas e Adaptadores), é como essa comunicação com o mundo exterior acontece. O hexágono (a aplicação) define "portas", que são essencialmente interfaces que especificam o que ela precisa ou o que ela oferece. Por exemplo, ela pode ter uma porta para "buscar dados de um produto" ou uma porta para "notificar o usuário". Essas portas são contratos, e são a única coisa que a aplicação conhece.

Do lado de fora do hexágono, temos os "adaptadores". Cada adaptador é uma peça de tecnologia que se conecta a uma porta. Para a porta "buscar dados de um produto", podemos ter um adaptador que implementa essa interface conversando com um banco de dados relacional. Mas, para os testes, podemos ter outro adaptador que usa um banco de dados em memória. Para a porta "notificar o usuário", podemos ter um adaptador que envia um e-mail, outro que envia um SMS ou um terceiro que simplesmente imprime no console. O ponto crucial é que a aplicação não conhece os adaptadores; ela só conhece as portas. Trocar um banco de dados ou adicionar uma nova forma de notificação se torna uma questão de "plugar" um novo adaptador, sem tocar em uma única linha de código no núcleo do sistema.

Dessa forma, podemos concluir que a Arquitetura Hexagonal não é apenas um padrão, mas uma filosofia de design que nos força a criar um limite claro entre o domínio do nosso problema e as tecnologias que usamos para resolvê-lo. Ela nos liberta da tirania da infraestrutura, permitindo que a aplicação evolua de forma independente e, mais importante, que seja testada de forma isolada e completa, sem a necessidade de subir um servidor web ou um banco de dados. Em um mundo onde as tecnologias mudam em um ritmo alucinante, a abordagem de Cockburn nos oferece um caminho para construir sistemas que são, ao mesmo tempo, resilientes, flexíveis e verdadeiramente focados em resolver os problemas de negócio para os quais foram criados.

Referências:

COCKBURN, Alistair. *Hexagonal architecture*. Alistair Cockburn's Web Site, 2005.

Autor da resenha: Rafael de Faria Neves Alves Franco