

**Jeffrey Palermo** 8h19 em 19 de agosto de 2013 Tags: arquitetura de cebola ( 4 )

## Onion Architecture: Parte 4 - Depois de quatro anos

Em 2008, criei um novo nome de padrão chamado Onion Architecture. Você pode ler as partes anteriores aqui: [parte 1](#) , [parte 2](#) , [parte 3](#) . [Ao longo desses quatro anos, falei sobre esse padrão em grupos de usuários, conferências e até foi publicado em um dos capítulos da ASP.NET MVC em Ação de Manning .](#)

Fiquei impressionado com a tração que este nome de padrão desfrutou. Pessoas de todo o país escreveram e falaram sobre o padrão. Alguns dos que eu notei estão aqui (por favor, comente com mais – eu aceito).

- A opinião de Ayende Rahien sobre Onion Architecture
- A visão “astuta” de Clinton
- Visualização e amostra de código de Tony Sneed
- Perguntas do StackOverflow sobre Onion Architecture
- Apresentação em vídeo do Grupo de Usuários .Net de Matt Hider sobre OA
- Uma pergunta interessante do StackOverflow.

Em 2008, defini quatro princípios da Onion Architecture:

- *O aplicativo é construído em torno de um modelo de objeto independente*
- *Camadas internas definem interfaces. Camadas externas implementam interfaces*
- *A direção do acoplamento é para o centro*
- *Todo o código principal do aplicativo pode ser compilado e executado separadamente da infraestrutura*

Embora tenha havido uma adoção significativa desse padrão, tenho recebido inúmeras perguntas sobre como implementá-lo em diversos ambientes. Na maioria das vezes, me perguntam sobre como isso se relaciona com o design orientado a domínio. Primeiro, a arquitetura onion funciona bem com e sem padrões DDD. Funciona bem com CQRS, formulários sobre dados e DDD. É apenas um padrão de arquitetura em que o modelo de objeto central é representado de uma maneira que não aceita dependências de código menos estável.

CodeCampServer foi uma amostra original da arquitetura onion, mas também cresceu como uma amostra de como fazer ASP.NET MVC de várias maneiras, como usar Portable Areas e como usar recursos MvcContrib como construtores de entrada. Se você está apenas procurando por arquitetura de cebola, tem muita coisa acontecendo. Eu empurrei uma solução muito mais simples que representa os conceitos de arquitetura onion. Eu intencionalmente não incluí um formulário de entrada de interface do usuário ou um contêiner IoC, que a maioria das pessoas associa à arquitetura onion. A arquitetura Onion funciona muito bem

sem os gostos de StructureMap ou Castle Windsor. Por favor, confira o código aqui e deixe-me saber se isso apresenta uma abordagem simples – esse é o objetivo.

Quando houver interesse suficiente, continuarei esta série com mais partes. O CQRS definitivamente merece algum endereçamento dentro dessa arquitetura, assim como os modelos de objeto que suportam UIs baseadas em tarefas.

Obtenha o código aqui no meu repositório BitBucket .