

**16:25**

Slide 3 - O que é

É um padrão arquitetural proposto por Robert Martin, mais conhecido como Uncle Bob. Que tenta fornecer uma metodologia a ser utilizada na codificação.

Slide 4 - Porque Utilizar

Algo que vale lembrar é que muitas vezes o pessoal se engana é que a arquitetura limpa não vai tornar o programador mais produtivo.

A fim de facilitar o desenvolvimento do código, permitir uma melhor manutenção, atualização e menos dependências.

A arquitetura contribui para que você tenha um código uma alta coesão (princípio da responsabilidade única) e um baixo acoplamento (quando uma classe depende de outra para funcionar) mas não só porque está utilizando a arquitetura quer dizer isso, temos que ficar atento na hora de implementar/manutenção e ver se não estamos delegando várias responsabilidades para uma mesma classe.

Slide 5 Como funciona

Uma arquitetura em camadas.

**Entidades** => São classes comuns a vários sistemas da empresa.

Vamos supor que tenho um sistema médico. Todo o sistema tem que lidar com classes como Médicos, Pacientes, Visitantes etc. Essas classes são chamadas de Entidades.

Além desses dados, as entidades podem implementar regras de negócios genéricas. Por exemplo, todo Médico tem uma especialidade.

Sendo assim nenhuma mudança no operacional do aplicativo pode afetar essa camada de Entidade e se alterarmos essa camada teremos que alterar toda a aplicação.

**Use Cases** => Nessa camada contém regras de negócios um pouco menos genéricas e normalmente relativas a um único sistema. Esses casos de uso orquestram o fluxo de dados e direcionam essas entidades a usarem suas regras de negócio em toda a empresa.

No nosso exemplo, se tivermos uma classe RegistroVisitantes que tem uma lista de objetos do tipo Visitante eu posso ter uma regra que define que cada visitante pode entrar somente para visitar um paciente específico. Também poderia ter regras para inserir ou deletar um Visitante, por exemplo.

“Não esperamos que as mudanças nesta camada afetem as entidades. Esperamos que as mudanças na operação do aplicativo afetem os casos de uso, portanto o software nesta camada.”

**Adaptadores** => Converter dados de um formato para outro. Qualquer adaptador necessário para converter dados de algum formulário externo, como um serviço externo para o formulário interno usado pelos casos de uso e entidades.

Por exemplo, se um sistema usa uma API Rest para comunicação com os clientes. As classes adaptadores são responsáveis por implementar os endpoints REST da API. Devem receber requisições e encaminhá-las para os casos de uso correspondentes. e também ao contrário receber os resultados retornados pelos casos de uso e convertê-los em documentos JSON que serão enviados para os clientes.

**Frameworks and Drives** => Geralmente não escrevemos muito código nesta camada. É a camada mais externa geralmente composta de estruturas e ferramentas como Banco de Dados, Estrutura da WEB etc...mantemos essas coisas do lado de fora, onde podem causar poucos danos.

**16:35**

Code

**17:45**