

Universidade Rural Federal de Pernambuco – UFRPE

Desenvolvimento em Camadas

Autora:

Fernanda Monteiro

29/05/2012 Recife – PE

Sumário

Arquitetura em camadas.....	3
1. Introdução	3
2. Estrutura de camadas	3
2.1 Camada de negócio	3
2.2 Camada de dados.....	3
2.3 Camada de comunicação	3
2.4 Camada de apresentação.....	4
3. Desenvolvimento em 3 camadas	4
4. Vantagens	5
5. Desvantagem	5

Arquitetura em camadas

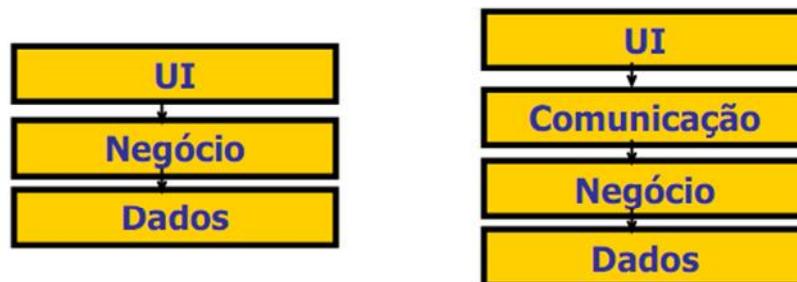
1. Introdução

Uma das grandes características de uma boa arquitetura no desenvolvimento de software é a divisão da aplicação em camadas.

A divisão em camadas pode proporcionar ao software um alto grau de flexibilidade, alta coesão, baixo acoplamento e com isso teremos o reuso dos componentes do software.

Cada camada é formada por um conjunto de classes com um determinado propósito. Este conjunto preocupa-se com detalhes específicos que serão 'escondidos' das camadas superiores.

2. Estrutura de camadas



2.1 Camada de negócio

Responsável por implementar a lógica do negócio. Nela, devem ficar as classes básicas de negócio, coleções de negócio e fachada do sistema.

2.2 Camada de dados

Responsável pela manipulação da estrutura de armazenamento de dados – executando inclusões, remoções, atualizações e consultas. Isola o resto do sistema do mecanismo de persistência utilizado.

2.3 Camada de comunicação

Responsável pela distribuição do sistema em várias máquinas.

2.4 Camada de apresentação

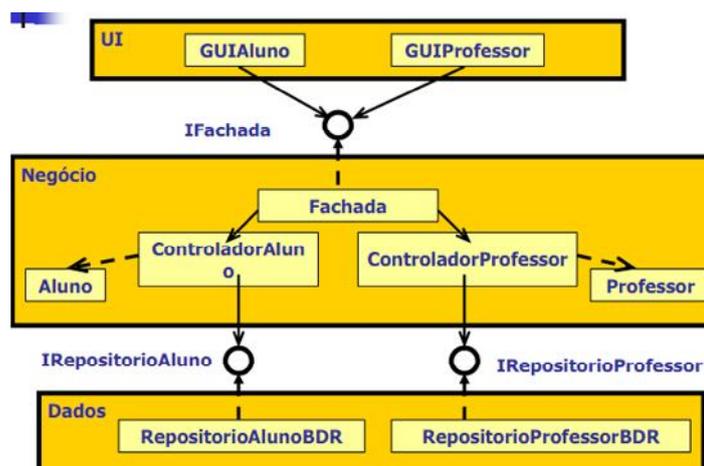
Esta camada é responsável por conter as interfaces do usuário e as regras inerentes a qualquer componente visual, regras de validação de input de dados e formatação de dados. Em uma aplicação que disponibilize serviços Web (Web Services), estes serviços também pertencerão à camada de apresentação, pois eles são considerados interfaces entre os sistemas.

Exemplo de diferentes configurações do padrão de arquitetura em camadas usando a tecnologia Java.



3. Desenvolvimento em 3 camadas

Quando falamos em desenvolvimento em 3 camadas, estamos nos referindo a um modelo de programação que prevê a divisão do programa em 3 partes bem definidas e distintas: interface, regras de negócio e banco dos dados.



No exemplo acima, podemos perceber que:

- Entre as camadas UI e Negócio há uma interface que uma classe `Fachada` do sistema implementará.

- ii. A classe Fachada é utilizada para oferecer um caminho único para acesso aos serviços da camada de regras de negócio.
- iii. As classes da UI, portanto, comunicam-se apenas com a classe Fachada, que por sua vez colabora com as outras classes internas da camada de regras de negócio para oferecer os serviços.
- iv. O Controlador pode conter regras de controle do sistema e delega ações da fachada para a camada de acesso a dados.
- v. Entre as camadas Negócio e Dados há sempre uma interface que uma classe Repositório implementará.
- vi. O Repositório armazena os objetos persistentes do sistema em algum meio de armazenamento físico (banco de dados, arquivo, etc.).

4. Vantagens

Promove separação de código relativo à interface com o usuário (GUI), comunicação, negócio e dados. Além disso, permite a mudança de implementação de uma camada sem afetar a outra, desde que a interface entre as mesmas seja mantida.

5. Desvantagem

Aumento no número de classes existentes no sistema.