

Descrição da Arquitetura do Software

Gabriel Alves Castro
Yuri Serka do Carmo Rodrigues
Vinicius Menezes Toshiyuki

19 de junho de 2019

Universidade de Brasília
yserka@hotmail.com
gabriel_alvesc1@hotmail.com
vtmsugimoto@gmail.com

1 Arquitetura Escolhida

A arquitetura de software escolhida como inspiração foi o Model-View-Controller(MVC), nesta arquitetura temos 3 camadas muito bem definidas sendo elas:

- **Model** onde é geralmente especificado quais são as entidades envolvidas no sistema juntamente com a conexão ao banco de Dados. Nesta camada que também estão implementadas as regras de negócio.
- **View** onde é geralmente feita a apresentação das opções para o usuário, ou seja, são as telas interativas. O foco desta camada é a apresentação de texto, sem qualquer lógica de negócio ou conexão com o banco de dados.
- **Controller** onde geralmente ocorre a validação das requisições dos usuários de acordo com as regras de autenticação, é nesta camada também onde ocorre a principal interação entre a camada **View** e **Model**, pode-se dizer que o controller é o intermediário entre eles, pois recebe e/ou envia dados para o Model e envia dados para a View.

O MVC é como a clássica programação orientada a objetos, ou seja, criar objetos que escondem as suas informações e como elas são manipuladas e então apresentar apenas uma simples interface para o mundo.

Entre as diversas vantagens do padrão MVC estão a possibilidade de reescrita da interface de usuário ou das Controladoras sem alterar o modelo, reutilização da IU para diferentes aplicações com pouco esforço, facilidade na manutenção e adição de recursos, bem como o reaproveitamento de código e facilidade de manter o código sempre limpo

A Figura abaixo mostra de forma geral como é a estrutura do MVC:

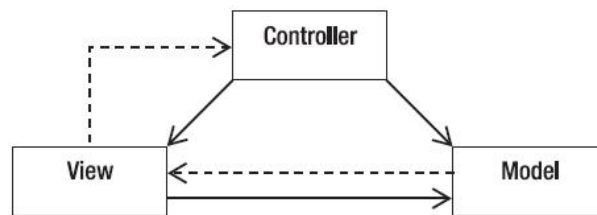


Figura 1: Arquitetura MVC

2 Aplicação no Sistema Desenvolvido

Neste Sistema a camada View está sendo apresentada no terminal, o que não é muito comum dado que esta arquitetura é bastante usada em sistemas Web, as controladoras estão responsáveis principalmente por apenas recuperar e mandar solicitação de alteração dos dados no banco de dados.

Na camada Model está especificado as estruturas de dados utilizadas, estas que representam cada entidade do sistema, e também as funções que fazem a conexão com o Banco de Dados. Vale ressaltar que não foi implementado uma camada de persistência, então quando refere-se ao banco de dados está sendo falado de *maps* que são coleções desordenadas de dados e que suportam adição, remoção e atualização.

A primeira escolha havia sido a arquitetura multicamadas, mas como a linguagem de programação escolhida não possui suporte eficiente para o padrão de orientação a objetos decidimos por ir pelo MVC. Apesar das duas arquiteturas serem parecidas, ainda há algumas nuances, como por exemplo no MVC é possível a comunicação direta entre todas as camadas do sistema, já em multicamadas não é aconselhável tal decisão.

Optamos também por esta arquitetura para treinarmos mais padrões diferentes e obter mais conhecimento sobre desenvolvimento de software.

3 Resultados Obtidos com a Abordagem

Podemos notar uma maior facilidade na manutenção das componentes do sistema, pois como cada camada já é especializada em realizar ações de seu domínio, o processo de refatoração e debug se tornaram mais objetivos.

O encapsulamento também é uma grande vantagem nesta arquitetura, pois cada camada usa apenas o valor de retorno da outra, portanto mudanças internas não são notadas em camadas superiores, este fato contribui para um código mais limpo e fácil de se dar manutenção.