Centro Paula Souza

Etec Philadelpho Gouvêa Netto

Guilherme Aparecido Ferreira Piovezan

TEMA: MÉTODOS MVC (*Model-View-Controller*) E DAO (*Data Access Objetic)*

São José do Rio Preto – São Paulo

2019

Sumário

[**Introdução ao MVC (*Model-View-Controller*)** 3](#_Toc7531457)

[ *Model* 3](#_Toc7531458)

[ *View* 3](#_Toc7531459)

[ *Controller* 3](#_Toc7531460)

[Exemplo 3](#_Toc7531461)

[**Introdução ao método DAO (*Data Access Object*)** 4](#_Toc7531462)

[Padrão do método DAO 4](#_Toc7531463)

[Implementação do método DAO 4](#_Toc7531464)

[Referências 5](#_Toc7531465)

# **Introdução ao MVC (*Model-View-Controller*)**

O *Model View Controller*, é um método de desenvolvimento, onde separamos o *Software* em três camadas básicas: Intermediação (“Ponte”), Controle e a Interface. Com o surgimento na déc. 70, pelo engenheiro Christopher Alexander, por um projeto onde iria resolver problemas específicos, logo alguns desenvolvedores de *softwares* abrangeram essa técnica a alguns projetos. A ideia inicial, ganhou notoriedade e grandes avanços surgiram, dando-se em um padrão de arquitetura de projetos dividido em três camadas.

## *Model*

O *Model* é uma camada responsável pela “leitura e escrita” de dados, ou seja, essa camada se comunica com o banco de dados da aplicação.

## *View*

O *View,* como o próprio nome sugere,é a camada responsável pela interface que na qual exibirá os dados do sistema e o seu conteúdo desenvolvido.

## *Controller*

O *Controller*, é a camada responsável pela interface e banco de dados – assemelhando-se a uma “ponte” – onde ela irá realizar todas as requisições de dados e o controle das outras camadas.

## Exemplo

Veja um exemplo deste tipo de arquitetura: imagine um sistema onde realiza o cadastro de clientes, para realizar a venda de produtos, na interface onde realiza o cadastro, existe o botão cadastrar, automaticamente clicando nele, ele acionará as classes de controle de dados e logo realizará a comunicação com o banco de dados para armazenar as informações, logo poderá ser feito a venda a este cliente, com seus dados cadastrados no sistema.

# **Introdução ao método DAO (*Data Access Object*)**

Toda a aplicação empresarial, necessita de interações com os *data source*,um banco de dados relacional, ou qualquer interação com tecnologia externa. Quaisquer interaçõespara realizar operações do tipo CRUD (*create, retrieve, update e delete*), podendo assim criar, recuperar, alterar e deletar qualquer objeto da aplicação. O principal objetivo do *Data Access Object* (DAO) é “encapsular[[1]](#footnote-1)” o acesso ao banco de dados, possibilitando que outras camadas da aplicação possam se comunicar com o mesmo.

## Padrão do método DAO

De acordo com o site DEVMIDIA, o padrão DAO, conforme citado pelo livro: ‘Core *J2EE Patterns’*,como: "o padrão utilizado para abstrair e encapsular todos os acessos ao data source. O DAO gerencia a conexão com o data source para obter e armazenar informações."

## Implementação do método DAO

A implementação do padrão envolve vários componentes, tais como sua interface, a fábrica (ou *FACTORY*) e o DTO (*Data Transfer Object*), onde todos são opcionais.

O DTO (*Data Transfer Object*) é responsável pelo transporte de dados recuperados numa base de dados através de sua camada lógica. Como por exemplo: para conseguirmos transferir uma lista de objetos do usuário com suas respectivas informações, que foram recuperados de uma “Camada de Acesso a Dados” para uma camada “Web”, a camada de serviços seria responsável por transferir de um DAO para um DTO os dados solicitados.

# Referências

Informações do site: ‘TABLELESS’ < https://tableless.com.br/mvc-afinal-e-o-que/> Acessado em 29 de abril de 2019 às 21:47.

Informações do site: ‘DEVMIDIA’ < https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308> Acessado em 29 de abril de 2019 às 22:01.

Informações do site: ‘DEVMIDIA’ < https://www.devmedia.com.br/implementando-o-data-access-object-no-java-ee/33339>Acessado em 30 de abril de 2019 às 15:45

1. Encapsular: incluir ou proteger algo em cápsula. [↑](#footnote-ref-1)