UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ENGENHARIA DE SOFTWARE

SÁVIO DE OLIVEIRA CAMACAM

PESQUISA SOBRE O PADRÃO ARQUITETURAL MVC

CAMPO MOURÃO

2016

1. INTRODUÇÃO

No estudo de Engenharia de Software somos apresentados ao conceito de Estilos Arquitetônicos, o que lida como os componentes de um projeto de software se relacionam, seguindo uma estrutura modularizada que permita que as partes sejam independentes e bem acopladas.

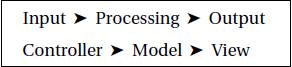
Alguns Estilos Arquitetônicos estão classificados em: Estilos Modulares, que trata da organização do código fonte; Componentes-conectores que tem uma visão de unidades de computação e seus conectores, que tem como foco o comportamento em tempo de execução; e também, Estilos de Alocação que tem a visão de integração de domínio, voltada para implantação do software.

Nessa pesquisa, foi estudado o Padrão MVC – Model-View-Controller – que é uma forma de dividir uma aplicação em três camadas básicas, onde cada uma tem suas características e funções bem definidas.

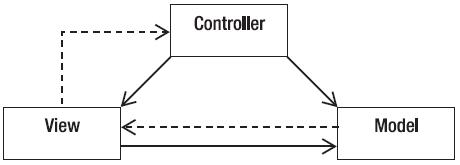
Com o aumento na complexidade dos sistemas/aplicações desenvolvidas hoje, essa arquitetura permite que, com qualquer tipo de alteração em uma das camadas, as demais terão impacto quase nulo, o que facilita a atualização de layouts, alteração nas regras de negócio e adição de novos recursos.

1. O PADRÃO ARQUITETURAL

O MVC inicialmente foi desenvolvido no intuito de mapear o método tradicional de entrada, processamento, e saída que os diversos programas baseados em GUI utilizavam. No padrão MVC, teríamos então o mapeamento de cada uma dessas três partes para o padrão MVC conforme ilustra a imagem abaixo [DEVMEDIA]:

**Figura 1**: Mapeamento das três partes de uma aplicação para o MVC.

A figura abaixo mostra uma organização baseada na captura de entradas do usuário, uma modelagem de regras isolada e o perfil visual do sistema para o usuário, gerenciados pelos objetos Modelo (Model), Visão (View) e Controlador (Controller).

**Figura 2**: Objetos utilizados no MVC e suas interações.

Uma das características de um padrão de projeto é poder aplicá-lo em sistemas distintos. O padrão MVC pode ser utilizado em vários tipos de projetos como, por exemplo, desktop, web e mobile [PROTOCOLOTI].

A seguir, veremos por partes o que cada camada desse padrão de projeto representa dentro da aplicação.

* 1. MODEL

O Model (modelo) é o objeto responsável pela manipulação e abstração da regra de negócio, com o uso de informações de forma mais detalhada, sendo recomendado que sejam feitos nele, consultas e cálculos e todas as regras do sistema/aplicação. É o Model que tem acesso a toda e qualquer informação do sistema, seja através de bancos de dados ou outros meios de armazenamento.

* 1. VIEW

A View é responsável por tudo que usuário final visualiza no sistema, com o desenho de interfaces, exposição de informações independente da fonte de origem. Um componente View renderiza o conteúdo de uma forma particular do Modelo e encaminha para o Controller as ações do usuário. Ele também acessa os dados do Model via Controller.

* 1. CONTROLLER

O Controller, com o auxílio da View e do Model, dá vida ao sistema, definindo seu comportamento interpretando as ações do usuário e as mapeando para chamadas do Model. EM um cliente de aplicações WEB, por exemplo, essas ações do usuário podem ser cliques em botões ou seleções de menus. Suas realizações também incluem ativar processos do negócio que levam a alteração do Model e com base em ações do usuário e nos resultados de processamentos, o Controller seleciona uma View a ser exibida como parte da resposta solicitada pelo usuário. Há normalmente um Controller para cada conjunto de funcionalidades relacionadas ao sistema.

1. EXEMPLO USANDO IONIC

Para exemplo nessa atividade, foi usado o Ionic Framework, uma ferramenta de desenvolvimento de aplicações mobile hibridas que geram uma aplicação instalável para Android e iOS.

A função do

1. REFERÊNCIAS BIBLIOGRÁFICAS

[DEVMEDIA] <http://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>

[PROTOCOLOTI] <http://protocoloti.blogspot.com.br/2012/12/mvc-model-view-controller.html>