OnMuseum

Projeto Arquitetural

There is guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute that allows you to toggle whether it is visible or hidden in this template. Use the Microsoft® Word® menu **Tools > Options > View > Hidden Text** check box to toggle this setting. There is also an option for printing: **Tools > Options > Print**.

# Objetivo

A arquitetura escolhida para ser utilizada na aplicação será MVC + Presentation Model. Presentation Model, descrito em detalhe por Martin Fowler em 2004 mas ainda marcado como “work in progress”, traz o estado da View para uma classe sem relação alguma com a renderização da UI. A ideia é que essa classe contenha todo estado da View, de maneira que a última possa ler esse estado e se renderizar por completo. Já a arquitetura MVC tem como objetivo básico separar a lógica de negócio da apresentação. A organização em camadas é a chave para a independência entre os componentes e alcança os objetivos como eficiência, escalabilidade, reutilização e facilidade de manutenção.

# Metas Arquiteturais e Filosofia

A arquitetura escolhida é uma junção das duas citadas acima e elas foram escolhidas embasada na possibilidade de eventuais alterações da apresentação e/ou lógica de negócio, sem a necessidade de alteração em todo o projeto. Outro ponto fundamental para a escolha dessa opção foi que o banco não será nativo e será necessária a utilização de recursos do dispositivo, como a câmera, e outras arquiteturas como Web-Mobile ou Nativa não supriria a necessidade da aplicação, uma vez que esta é híbrida.

# Premissas e Dependências

# Requisitos Críticos da Arquitetura

<http://martinfowler.com/eaaDev/PresentationModel.html>

<http://www.codeproject.com/Articles/685240/Using-MVC-to-Understand-ASP-NET>

# Decisões, Restrições e Justificativas

# Mecanismos Arquiteturais

O projeto será particionado da seguinte forma:

**Controller**: De posse do Presentation Model, pode atualizar a View e também tem acesso à camada de serviços (persistência) da aplicação. É composto também as Menssages suportadas que serão usadas pela View para repassar os eventos gerados pela iteração do usuário;

**View**: A Activity assume o papel de View; como ela é o “Entry Point” da aplicação, também fica responsável pela instanciação do Presentation Model, nesse momento a View se registra como Observer do Model. Também instancia o Controller, que recebe o Presentation Model como parâmetro do seu construtor. Isso tudo acontece no onCreate() onde normalmente é enviada uma Message de inicialização para o Presenter.

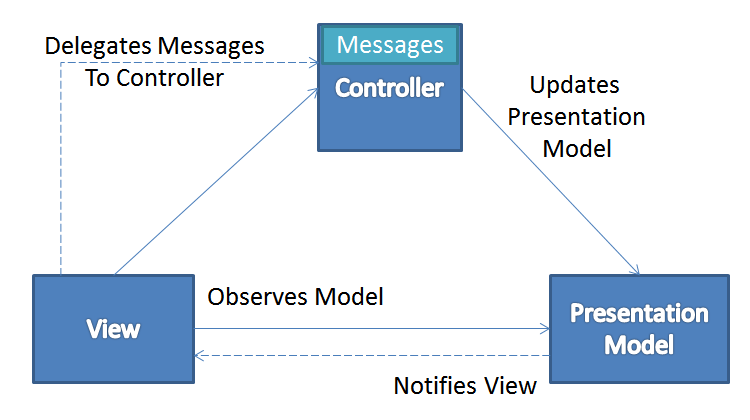
**Presentation Model**: É instanciado pela View e repassado para o Controller;

**DAO**: Relativo a persistência e/ou conexão com o banco

**Util**: Destinado a classes que não pertencem somente ao negócio, facilitando assim um possível reuso.

# Principais Abstrações

# Camadas do Framework da Arquitetura



# Visões Arquiteturais

• Lógica:

• Operacional:

• Caso de uso:

