# 4. Padrões de Desenvolvimento

## • Padrões de Codificação das Telas (pode estar também junto com cada respectiva interface)

## • Padrões de Documentação e Codificação

Para documentação do código JAVA e Flex, será utilizado o padrão JavaDoc.

Para as Classes utilizaremos o padrão exemplificado abaixo:

***/\*\****

***\* @description Classe para cadastro de Chamados***

***\* @author***

***\* @version 1.1***

***\* @since Release 01 da aplicação***

***\* Criado em : 16/08/2010***

***\*/***

Para os métodos utilizaremos o padrão demonstrado a seguir:

***/\*\****

***\* Descrição do método.***

***\*@param nomeParametro - descrição***

***\*@return retorno***

***\*/***

Para melhor visualização dos comentários durante o projeto, seguiu-se o seguinte padrão:

**/\***

**\* Para comentários que**

**\* apareçam em várias linhas**

**\*/**

**Ou**

**// e este caso seja comentado apenas uma linha**

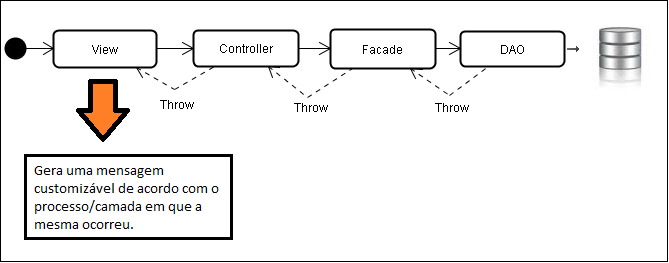
Para a codificação das Classes utilizaremos o padrão *Camel Case*, conforme exemplo abaixo:

*GrupoAtendimento*

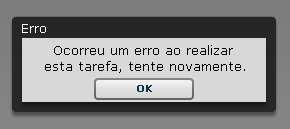
Para os método e variáveis a primeira palavra será em minúsculo, e caso haja mais de uma palavra estas deverão começar com a primeira letra em maiúsculo. Exemplo:

*calculaValorPrioridade()*

## • Padrões de Tratamento de Exceções e Erros

Como o sistema utiliza o padrão MVC, as exceções devem seguir o seguinte padrão:

Todos os processos de negócio devem respeitar este modelo de tratamento de exceções, para que fique genérico e de fácil manipulação das mensagens. A camada de **controle**  deve ser a responsável por buscar a mensagem **amigável** no repositório das mensagens do sistema. Quando a mensagem atingir a camada **View** ela será exibida utilizando o componente Alert, como no exemplo.



## • Padrão usado no Banco de Dados