

Projeto da Interface com o usuário

Introdução

O projeto de Interface com Usuário (IU) define como o usuário comandará o sistema e como o sistema apresentará as informações a ele.

Separação das camadas

- Um dos princípios fundamentais para um bom projeto de software é a separação da apresentação (camada de IU) da lógica de negócio (camada de LN). Essa separação é importante por diversas razões, dentre elas:
 - O projeto de IU e o projeto da LN tratam de **diferentes preocupações**. No primeiro, o foco está nos mecanismos de interação e em como dispor uma boa IU. O segundo concentra-se em conceitos e processos do negócio.
 - Usuários podem querer ver as **mesmas informações de diferentes maneiras** (usando diferentes interfaces, tais como interfaces ricas de sistemas desktop, interfaces de aplicações Web tradicionais, interfaces de linha de comando etc.). Neste contexto, separar a IU da LN permite o desenvolvimento de múltiplas apresentações.
 - **Objetos não visuais** são geralmente **mais fáceis de testar** do que objetos visuais. Ao separar objetos da LN de objetos de IU, é possível testar os primeiros sem envolver os últimos.

Camada de IU

- A camada de IU envolve dois tipos de funcionalidades:
 - **Visão:** refere-se aos objetos gráficos usados na interação com o usuário;
 - **Controle de Interação:** diz respeito ao controle da lógica da interface, envolvendo a ativação dos objetos gráficos (p.ex., abrir ou fechar uma janela, habilitar ou desabilitar um item de menu etc.) e o disparo de ações.

O Padrão Modelo-Visão-Controlador (MVC)

O padrão Modelo-Visão-Controlador (MVC) considera três papéis relacionados à interação humano-computador.

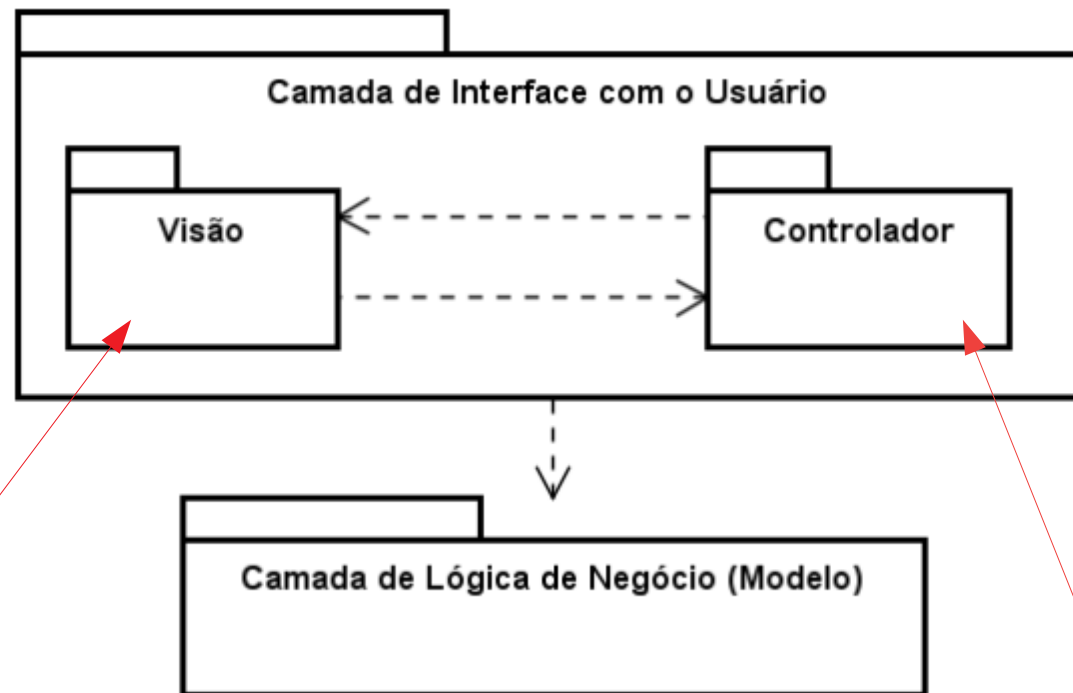
- O **modelo** refere-se aos objetos que representam alguma **informação sobre o negócio** e corresponde, de fato, a objetos da **camada de Lógica de Negócio**.
- A **visão** refere-se à entrada e à exibição de informações na **IU**.
- Qualquer requisição é tratada pelo terceiro papel: **o controlador**. Este pega a entrada do usuário, envia uma requisição para a camada de lógica de negócio, recebe sua resposta e solicita que a visão se atualize conforme apropriado.

Assim, a Interface com Usuário IU é uma combinação de visão e controlador (Atenção a essa informação – ver próximo slide)

Atenção ao CGT:

- Neste ponto é importante **distinguir os controladores do padrão MVC** das classes gerenciadoras de caso de uso do **Componente de Gerência de Tarefas (CGT)**.
- O componente de Gerência de Tarefas representa classes da lógica de negócio (lógica de aplicação), que encapsula e centraliza o tratamento de certos casos de uso.
- Um controlador do padrão MVC é um **controlador de interação**, ou seja, ele controla a lógica de interface, abrindo e fechando janelas, habilitando ou desabilitando botões, enviando requisições etc.

O Padrão Modelo-Visão-Controlador (MVC)

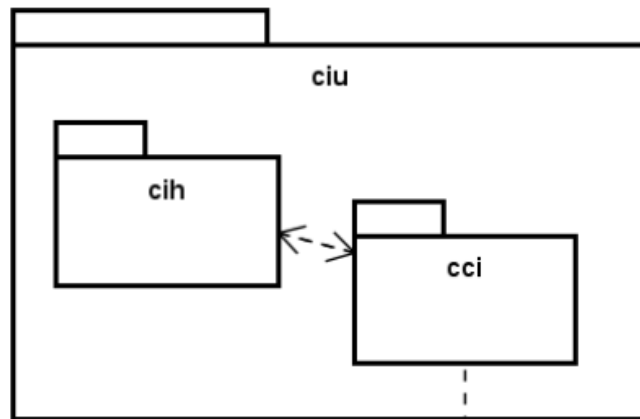


Componente de Interação Humana (cih)

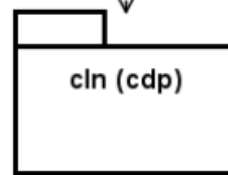
Componente de Controle de Interação (cci)

O Padrão Modelo-Visão-Controlador (MVC)

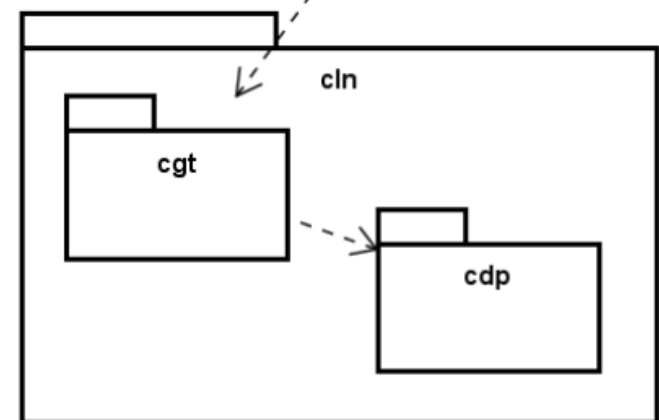
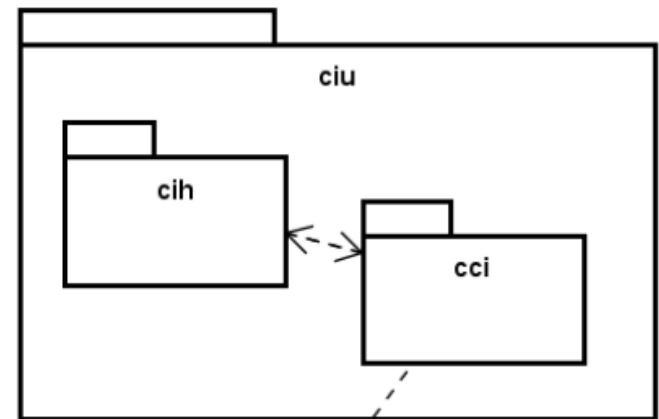
Interface com
o Usuário



Lógica de Negócio

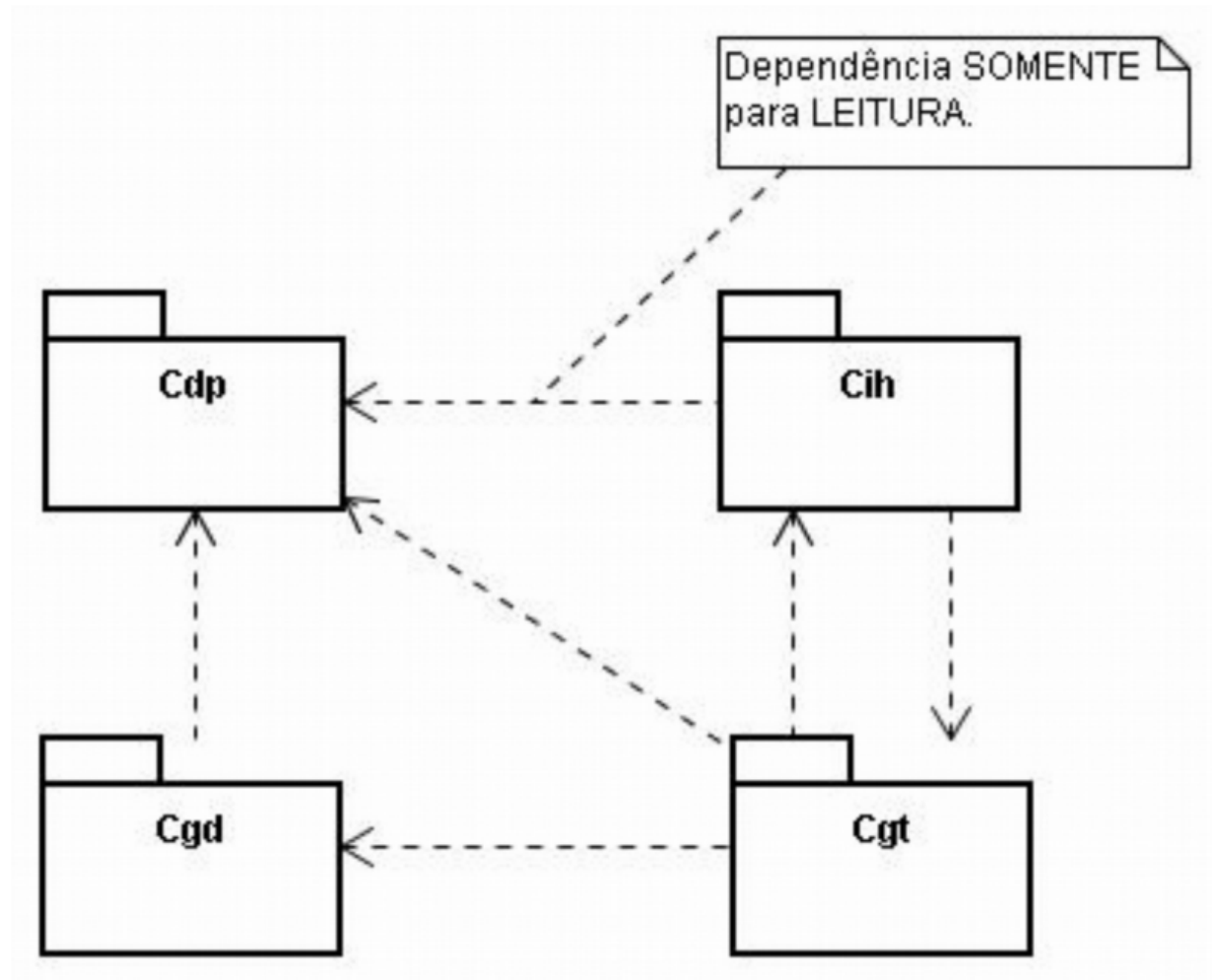


Padrão Modelo de Domínio

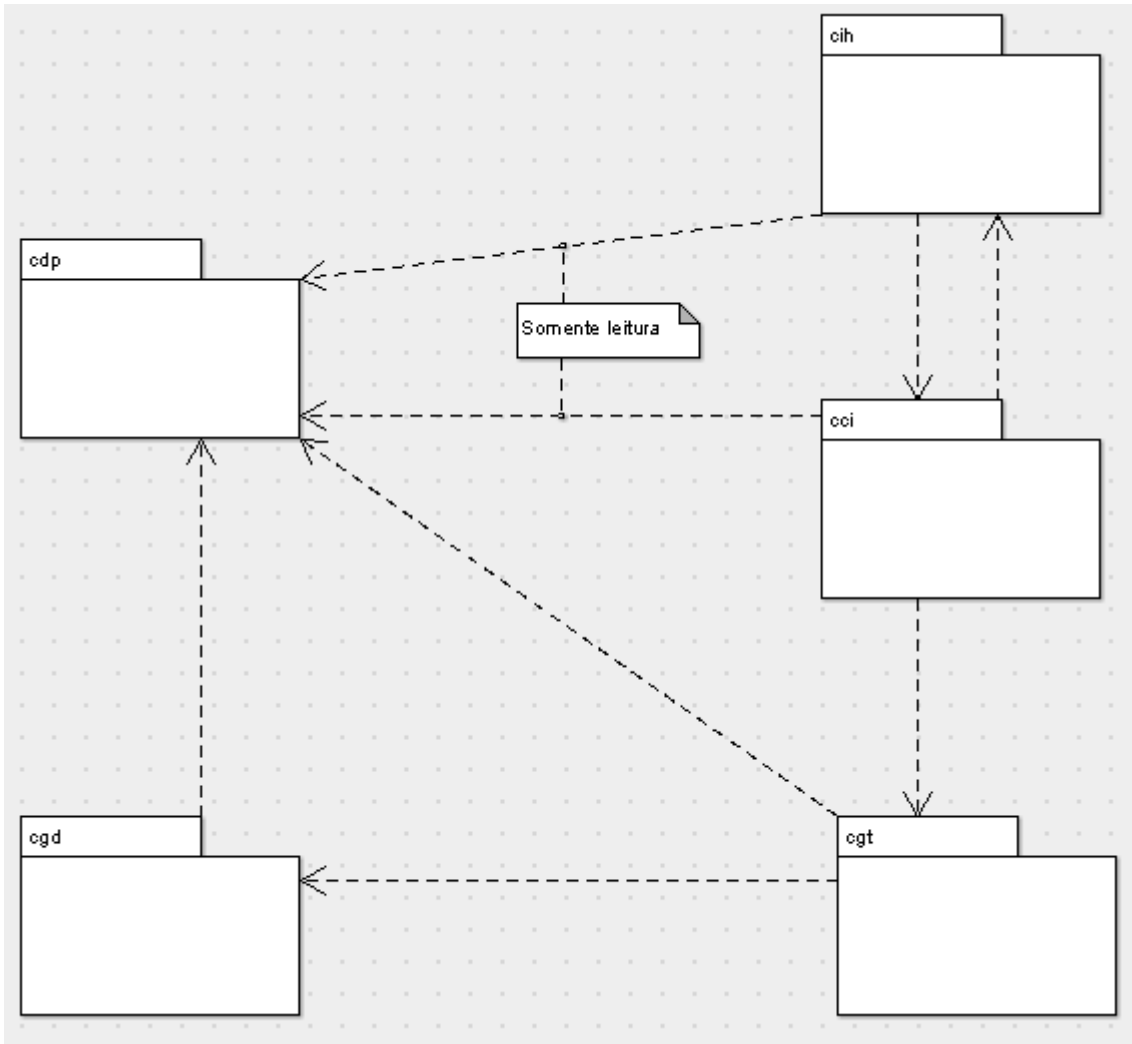


Padrão Camada de Serviço

Voltando ao Modelo de Coad e Yourdon



Organizando os pacotes e ajustando o nosso modelo



cdp: Componente/Camada de Domínio do problema

cgd: Componente/Camada de Gerência de Dados

cgt: Componente/Camada de Gerência de Tarefas

cci: Componente/Camada de Controle de Interação

cih: Componente/Camada de Interação Humana

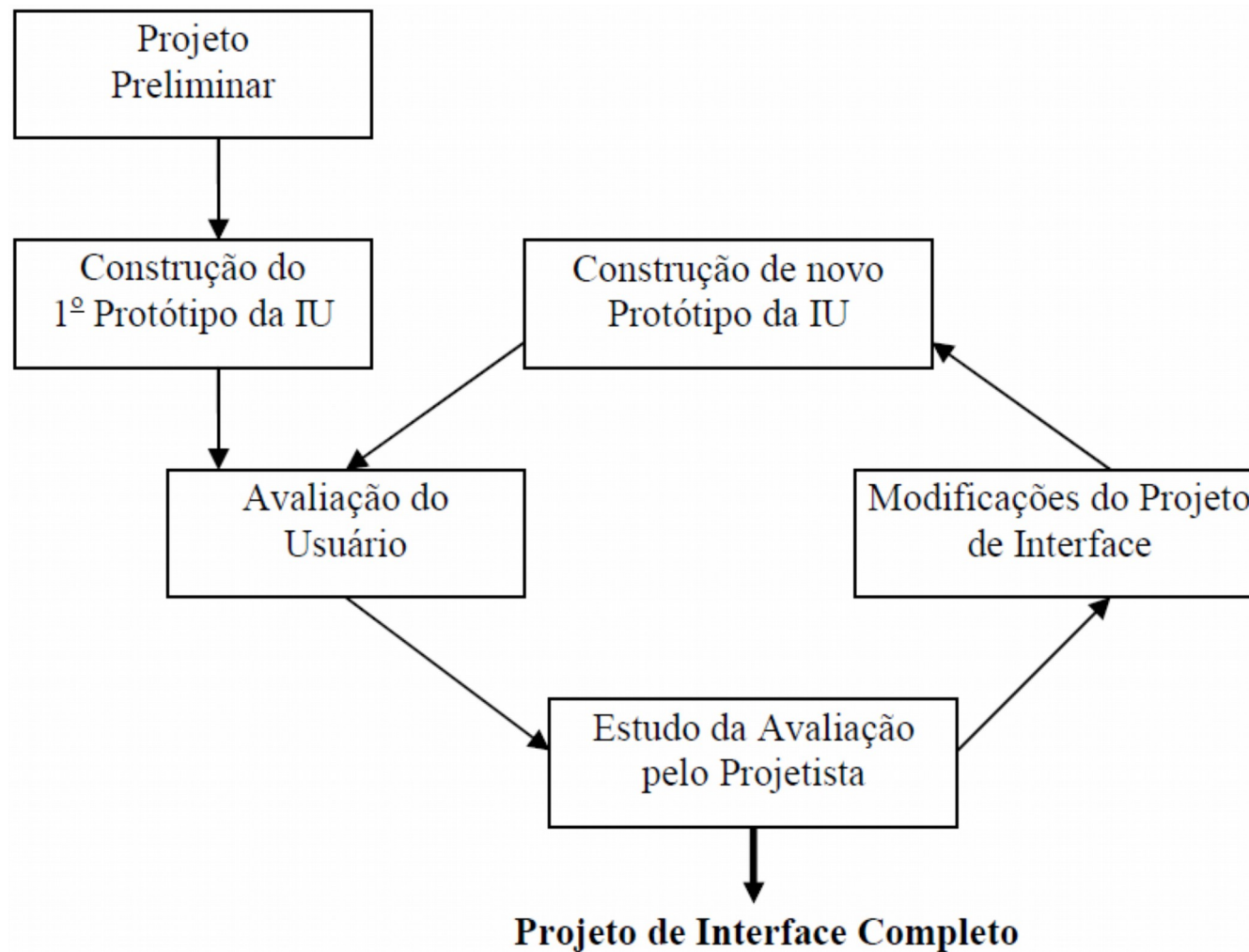
Projeto da Interface com o usuário

- Questões básicas:
 - Quem é o usuário?
 - Como ele pretende interagir com o sistema?
 - Como ele interpreta a informação produzida pelo sistema?
 - O que ele espera do sistema?
- Os casos de uso são a base para a construção do projeto de interface, pois agrupam informações dos usuários e as tarefas que eles vão desempenhar.

Passos para construção da interface

- 1) Definir as funcionalidades acessíveis a partir da IU (interface com o usuário) e o fluxo global da interação: caso de uso x interface adequada.
- 2) Estabelecer o perfil do usuário (Novato, Conhecedor Esporádico e Conhecedor Frequente).
- 3) Considerar princípios gerais de projeto de IU (Ajuda, mensagens de erros, tipos de comandos) de acordo com o perfil.
- 4) Construir protótipos (ver próximo slide).
- 5) Avaliar Resultado (visões qualitativas e quantitativas).

Construir Protótipos (prototipação)



Projeto da Visão (cih)

- A porção do sistema que lida com a visão da interface com o usuário (tratada pelo componente de interação humana) deve ser mantida tão independente e separada do resto da arquitetura do software quanto possível. Aspectos de interface com o usuário provavelmente serão alvo de alterações ao longo da vida do sistema e essas alterações devem ter um impacto mínimo nas demais partes do sistema.
- Com base nos casos de uso, deve-se projetar uma hierarquia de comandos, definindo barras de menus, menus pull-down, ícones etc., que levem à execução dos casos de uso, quando acionados pelo usuário.

Dica

Evite projetar interfaces com classes básicas disponibilizadas da interface. Crie suas classes a partir das existentes, facilitando assim a alterabilidade e manutenção.

Táticas de Usabilidade

- Facilidade de Ajuda

- quando a ajuda estará disponível e para que funções do sistema;
- como ativar (botão, tecla de função, menu);
- como representar (janela separada, local fixo da tela);
- como retornar à interação normal (botão, tecla de função);
- como estruturar a informação (estrutura plana, hierárquica, hipertexto)

- Mensagens de Erro e Avisos

- Descreva o problema de forma simples e objetiva
- Assistência a recuperação de erros
- Indicar consequências da ação
- Acompanhar a mensagem de dica visual ou sonora
- Não censure o usuário

Táticas de Usabilidade

- **Tipos de comandos:** Diferentes grupos de usuários têm diferentes necessidades de interação. É necessário definir:
 - se toda opção de menu terá um comando correspondente;
 - a forma do comando, tais como controle de sequência (p.ex., ^Q), teclas de função (p.ex., F1) e comandos digitados;
 - quão difícil é aprender e lembrar o comando;
 - possibilidade de customização de comandos (macros);
 - padrões para todo sistema e conformidade com outros padrões, tal como o definido pelo sistema operacional (Material Design, por exemplo) ou por produtos de software tipicamente utilizados pelos usuários.
- **Tempo de resposta:** em processos demorados é importante informar o usuário a progressão

Princípios Gerais

- 1) Seja Consistente
- 2) Ofereça retorno significativo (relação com princípios ágeis – descobrir as 20% mais importantes funcionalidades)
- 3) Peça confirmação das ações destrutivas (apagar dados pode ser fatal, o usuário DEVE ser alertado)
- 4) Permita reversão (apesar de nem sempre ser possível é desejável)
- 5) Reduza a informação que precisa ser memorizada entre as ações
- 6) Busque eficiência da interface
- 7) Trate possíveis erros do usuário (o bom uso de exceções é sempre interessante)
- 8) Classifique as atividades por função
- 9) Forneça ajuda sensível ao contexto
- 10) Use verbos ou frases curtas para nomear comandos

Diretrizes para a apresentação

- 1) Mostre apenas informações relevantes
- 2) Use formatos de apresentação que permitam assimilação rápida, tais como gráficos (em Android temos o GraphView, por exemplo) e figuras.
- 3) Use rótulos consistentes, abreviaturas e cores previsíveis.
- 4) Produza mensagens de erro significativas.
- 5) Projete adequadamente o layout de informações textuais. Letras maiúsculas e minúsculas, indentação, agrupamento de informações
- 6) Separe diferentes tipos de informação. Painéis podem ser usados para este fim (JPanel, por exemplo).
- 7) Use formas de representação análogas às do mundo real para facilitar a assimilação da informação. Para tal considere o uso de figuras, cores etc.

Projeto do Controle de Interação (cci)

- Visa definir as classes responsáveis por controlar a ativação/desativação de objetos de visão e enviar requisições a camada de negócio.
- Em sistemas desktop deve haver pelo menos 1 classe controladora.

Uso de cores

A cor pode ter um efeito positivo ou negativo ao comunicar ideias ao usuário

Cores quentes:

- Vermelho: perigo, sobressalto, alarme, detenção;
- Amarelo: precaução, atenção.
- Laranja: Ação, precaução.

Cores frias:

- Azul: calmo, repouso, silêncio, tranquilo
- Verde: disponibilidade, liberdade, proteção
- Violeta: duelo, espiritualidade, melancolia

Dúvidas?

