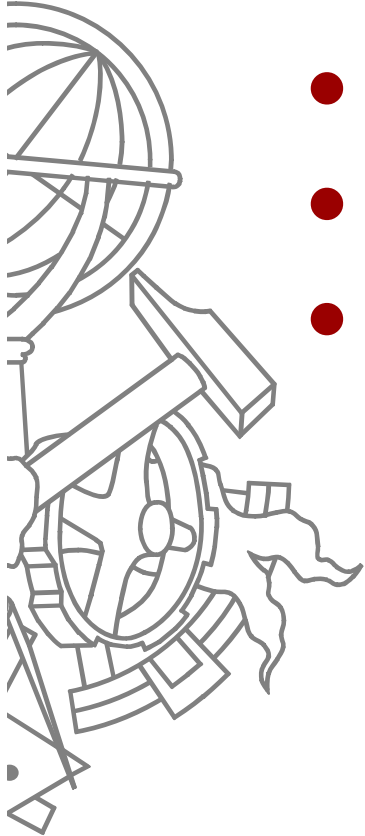


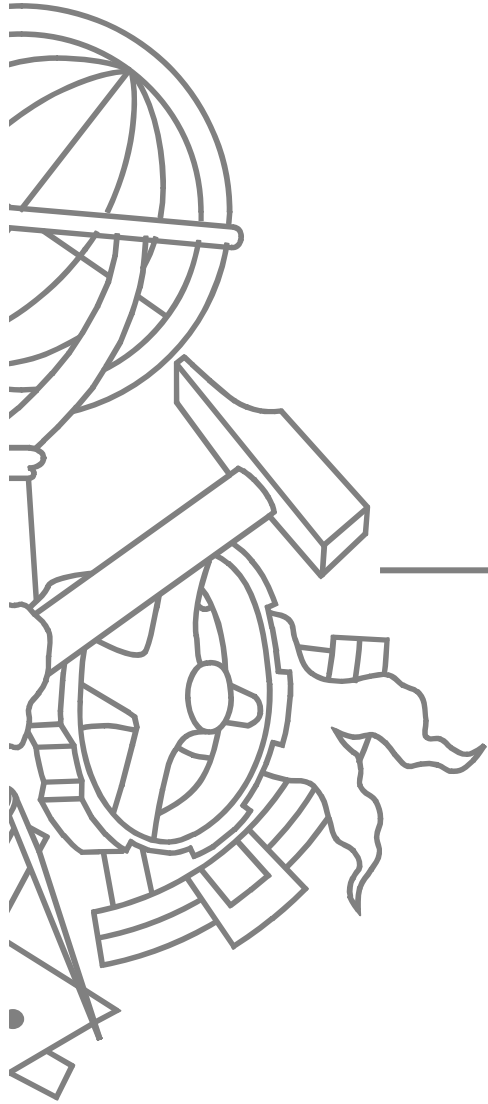
Lógica de Apresentação

Parte 2.5

Padrões de lógica de apresentação

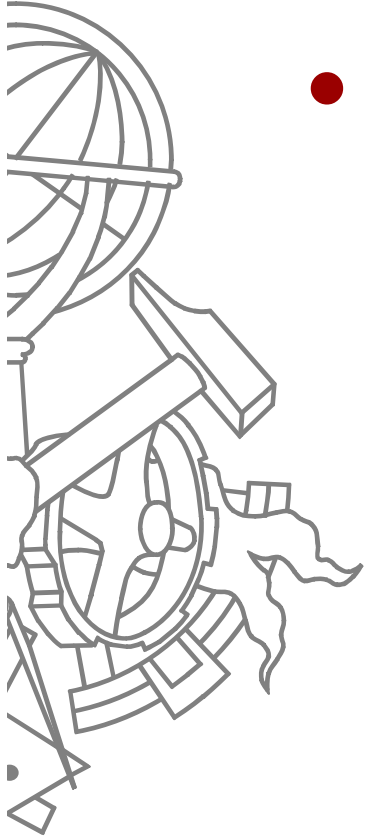


- Model-View-Controller [PoEAA] [ESP]
- Transform View [PoEAA]
- Two-step Transform [PoEAA]

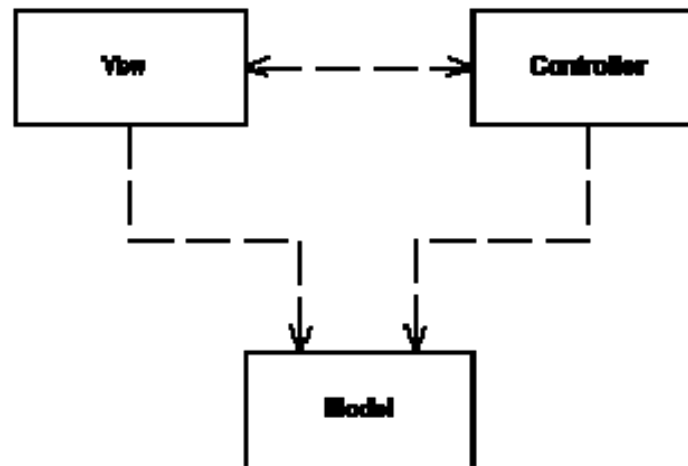


Model-View-Controller

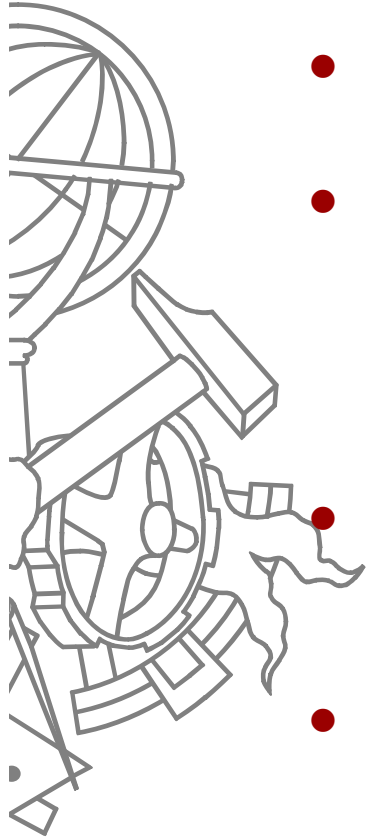
Model-View-Controller



- Splits user interface interaction into three distinct roles

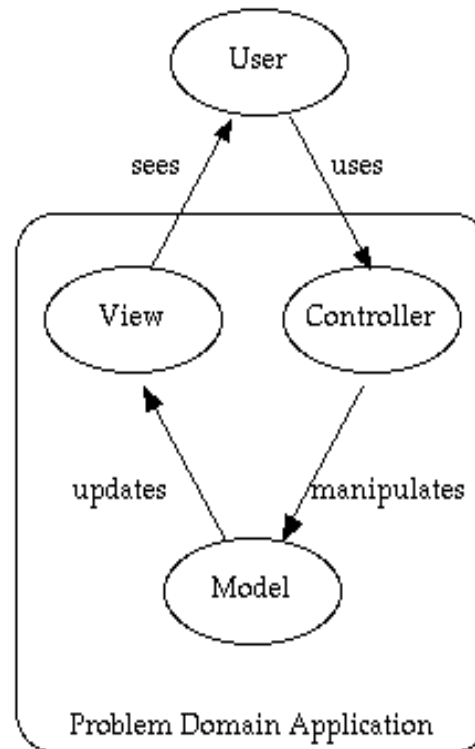
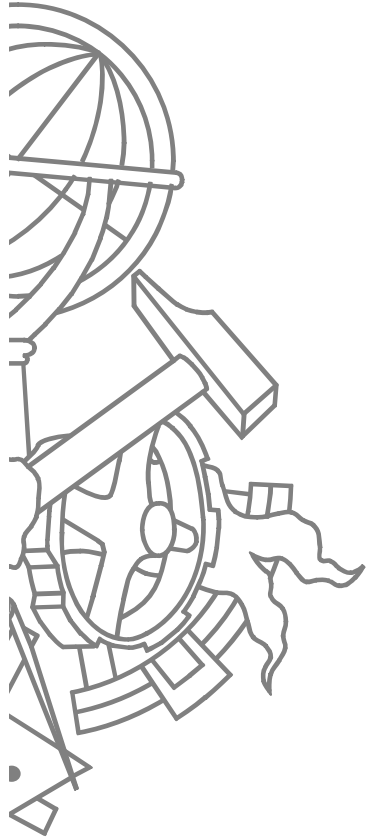


MVC

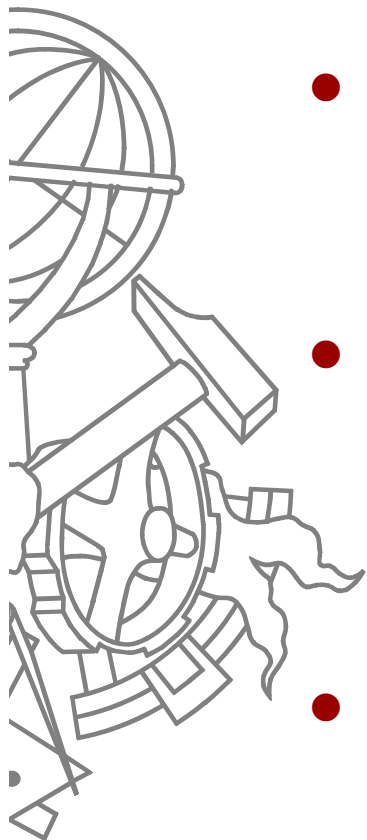


- Este padrão de desenho é simples mas bastante poderoso, baseando-se em:
- **Modelo**
 - O núcleo da aplicação, ou seja a informação relevante (apenas dados e estado). Quando uma alteração significativa acontece no **modelo**, as **vistas** são actualizadas
- **Controlador**
 - A interface com o utilizador através da qual se manipula o **modelo**.
- **Vista**
 - A interface com o utilizador que visualiza o **modelo**. Uma **vista** deve registar-se junto do **modelo** para receber as notificações de alterações.

MVC

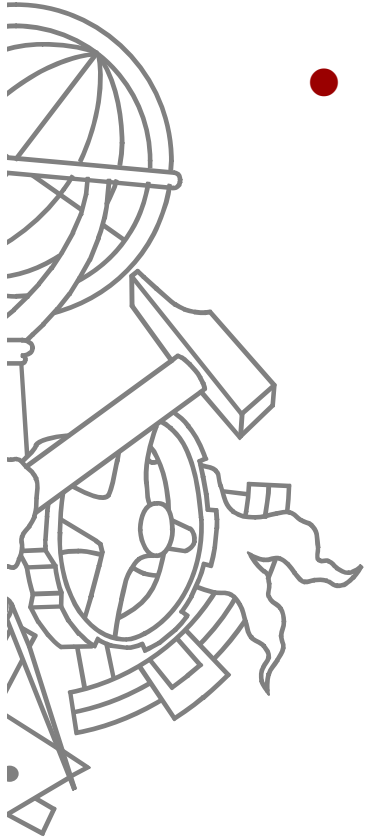


Modelo



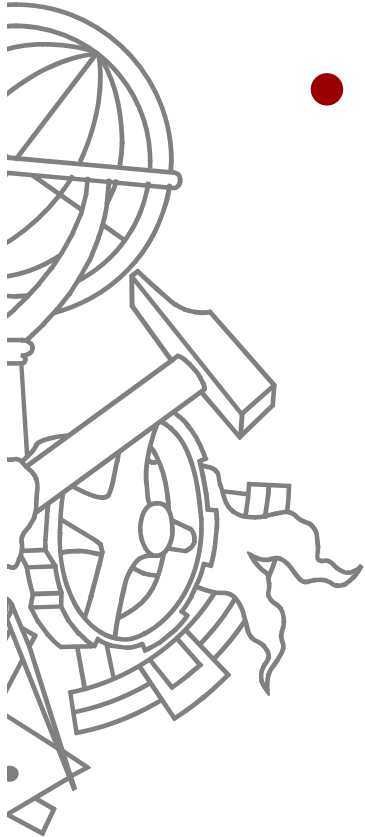
- O modelo deve ser completamente independente da sua representação externa.
 - Favorece a reutilização do modelo
- Operações específicas do modelo
 - *Accessors*
 - *Mutators*
 - etc
- As vistas e os controladores devem poder ser alterados ou substituídos sem impacto no modelo

Modelo



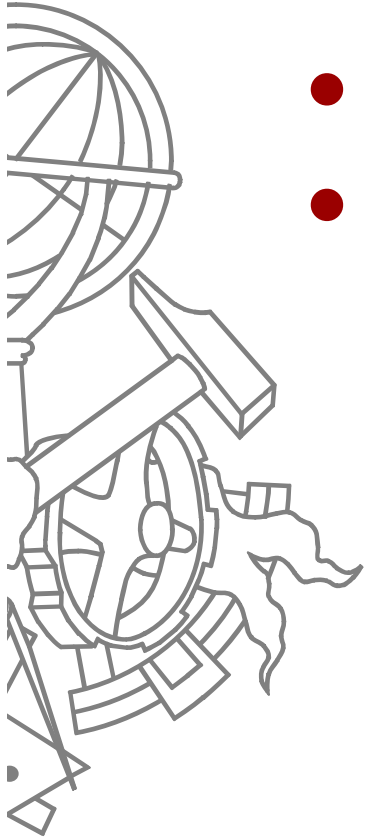
- Operações fundamentais MVC
 - adicionarVista()
 - Permite o registo de uma vista junto deste modelo por forma a receber notificações de alteração do modelo
 - notificarVistas()
 - Envia a todas as vistas registadas uma mensagem indicando que o modelo sofreu uma alteração, permitindo assim a actualização das vistas

Vista

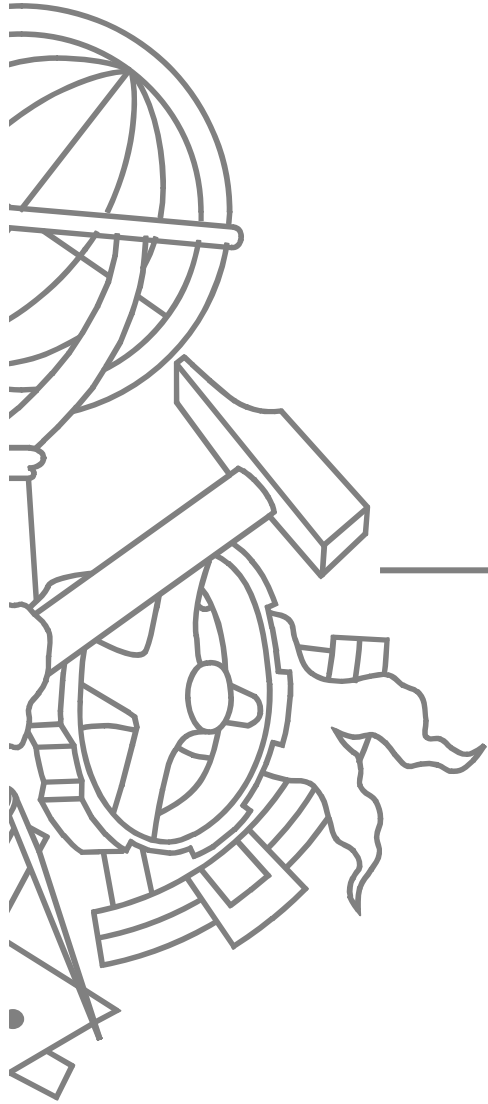


- Operação fundamental MVC
 - Update()
 - Enviado pelo modelo para indicar que houve uma alteração; dá à vista uma oportunidade para se actualizar de forma a reflectir o novo estado do modelo

Controlador



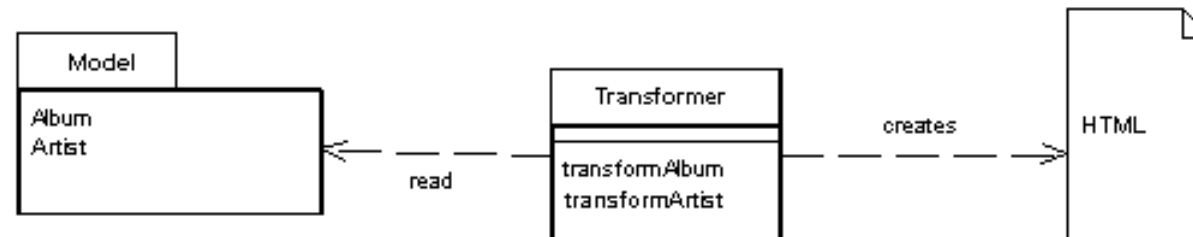
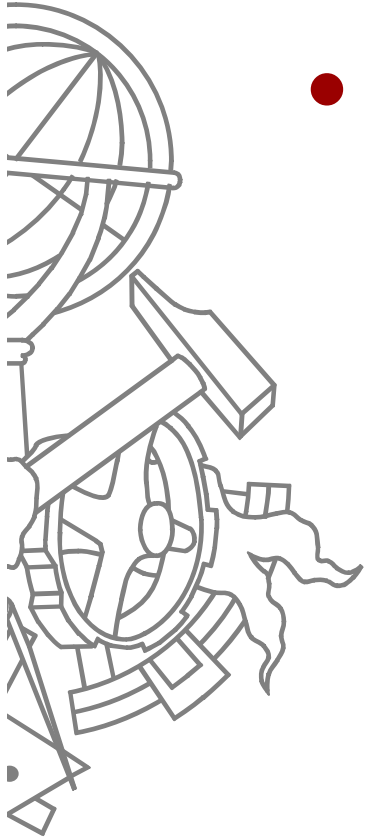
- Operações específicas do controlador
- Tem que ter acesso ao modelo para invocar os métodos correspondentes que permitem manipular o objecto em resposta aos pedidos do utilizador



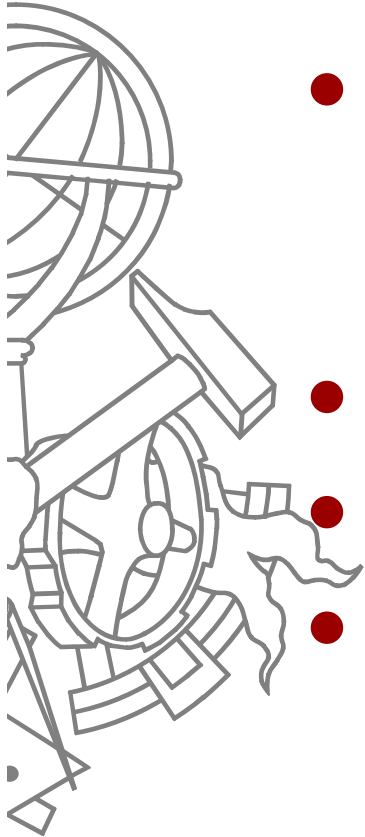
Transform View

Transform View

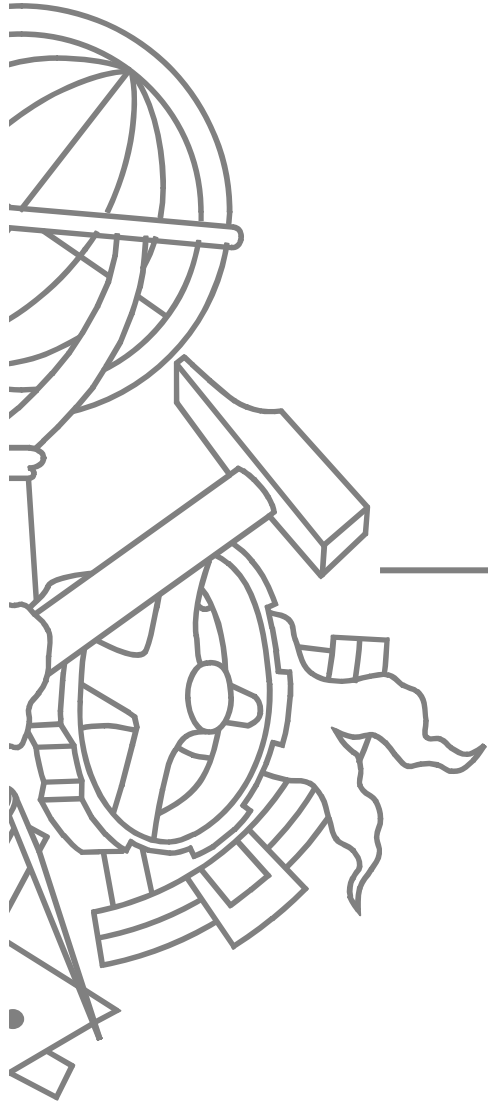
- A view that processes domain data element by element and transforms it into HTML



Transform View



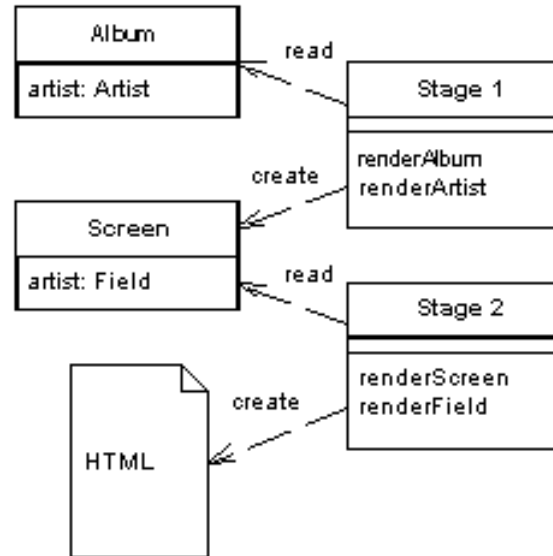
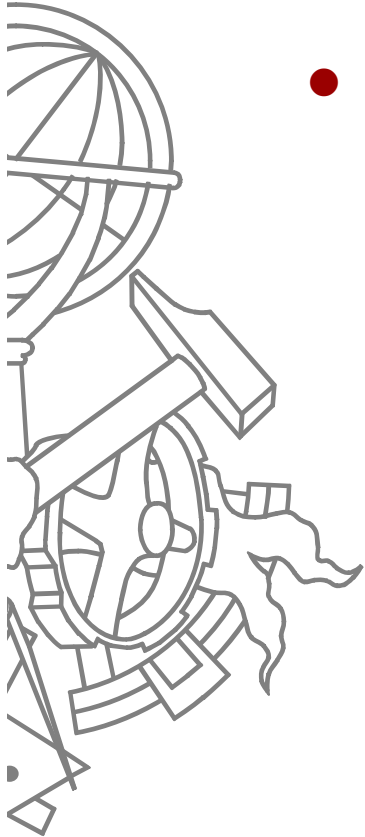
- Organizada em transformações separadas para cada tipo de elemento de entrada (objectos do domínio)
- XML (DOM) → XSLT → HTML
- Orientado aos documentos
- Independência das plataformas



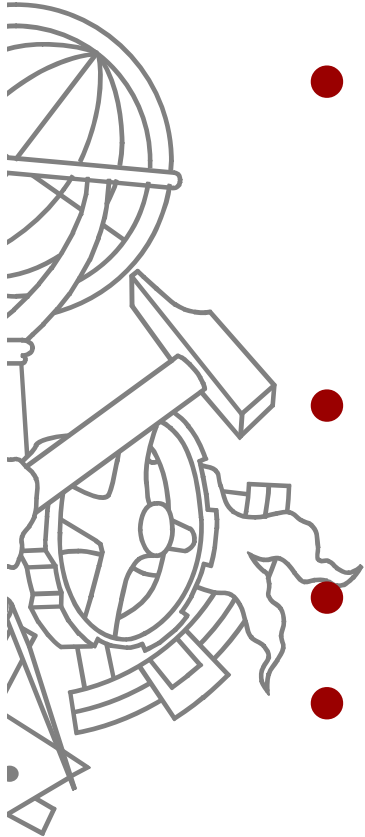
Two Step Transform

Two-step Transform

- Turns domain data into HTML in two steps: first by forming some kind of logical page, then rendering the logical page into HTML



Two-step Transform



- Transformação em dois passos:
 - Informação do domínio para apresentação lógica (sem formatação)
 - Formatação em HTML, XML, WAP, ...
- Facilita alterações globais e apresentação uniforme
- Facilita a implementação multi-canal
- Exemplos:
 - Duas transformações XSLT
 - JSP e Custom tags