Academic Advisor

Design

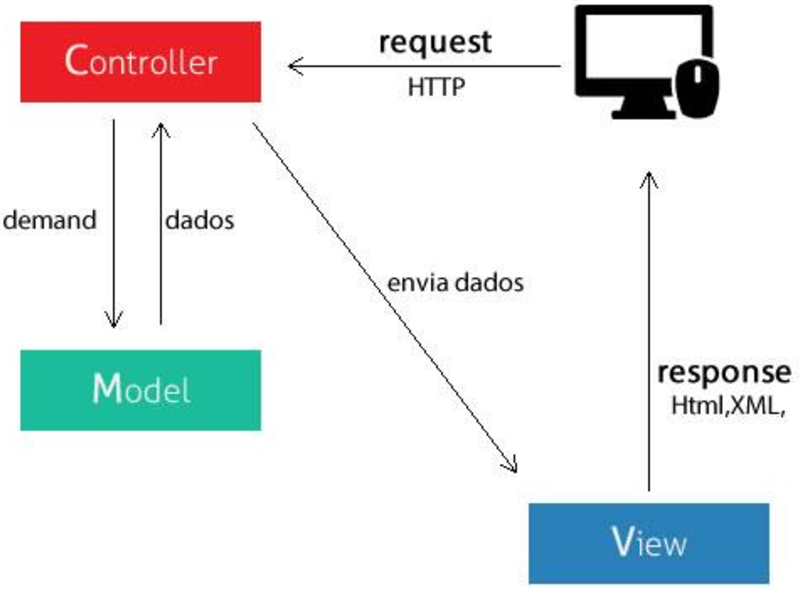
**Nome do padrão**

MVC (Model-View-Controller)

**Overview: descrição breve do padrão**

**Arquitetura de camada MVC**

O MVC (Model-View-Controller) é um padrão de projeto arquitetural baseado em outros padrões como o Observer, Composite e Strategy. O MVC, como o nome sugere, possibilita a separação de um projeto em múltiplas camadas, das quais fazem parte: Modelo (Model), Visão (View) e Controlador (Controller).



**Model -** É responsável pela leitura e escrita de dados, e também de suas validações.

**View -** Simples: a camada de interação com o usuário. Apenas faz a exibição dos dados (html ou xml).

**Controller -** Responsável por receber todas as requisições do usuário. Seus métodos chamados actions são responsáveis por uma página, controlando qual model usar e qual view será mostrado ao usuário.

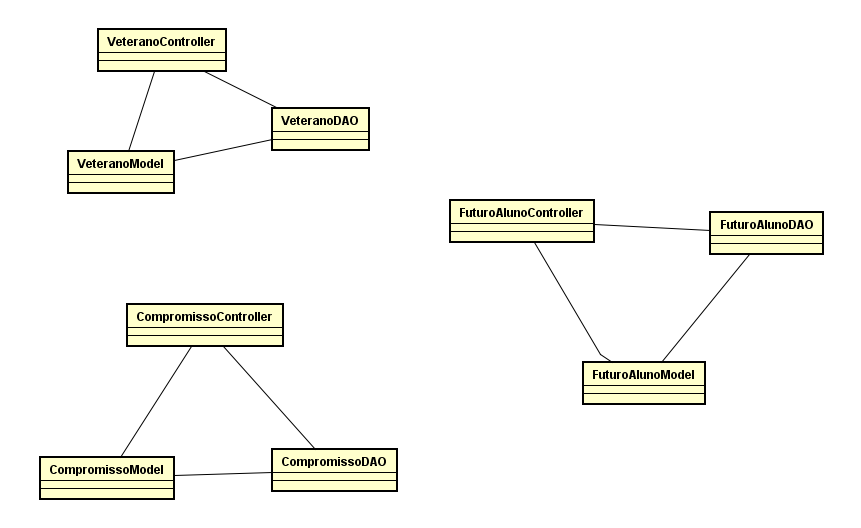
**DAO - Objeto de acesso a dados**

Objeto de acesso a dados (ou simplesmente DAO, acrônimo de Data Access Object), é um padrão para persistência de dados que permite separar regras de negócio das regras de acesso a banco de dados. Numa aplicação que utilize a arquitetura MVC, todas as funcionalidades de bancos de dados, tais como obter as conexões, mapear objetos Java para tipos de dados SQL ou executar comandos SQL, devem ser feitas por classes DAO.

**Structure: diagrama de classes do padrão utilizado na aplicação**

Os diagramas de classe representam o padrão de visão de participantes. Para cada caso de uso vai existir um diagrama com todos as classes de sistema envolvidos. Não se representam entidades de tela de usuário, no caso do MVC, os diagramas possuem as classes de controle, de modelo e DAO.

*Figura 01 – Exemplo de uma visão dos participantes de um cenário. Esse padrão se repete para outros casos de uso.*



**Behavior: diagrama de sequência do padrão utilizado na aplicação**

Não prevê perda de mensagens, pois é um sistema simples de cadastro. Os diagramas não representam integrações com outros sistemas. Também não se trata de algum sistema de software embarcado ou sistema de TI em redes.

**Elementos**

**Objetos:**

• **Fronteira (boundary)** – Classes de interface com o mundo externo (ex: GUI, sistemas externos)

**• Controle (control)** – Coordenam o comportamento do caso de uso definindo uma interface entre classes fronteira e entidade

• **Entidade (entity)** – Classes que armazenam informações manipuladas pelo sistema

**Atores:**

Aluno, Futuro-Aluno, Administrador;

**Mensagens síncronas:**

Não existe necessidade de mensagens assíncronas no sistema.

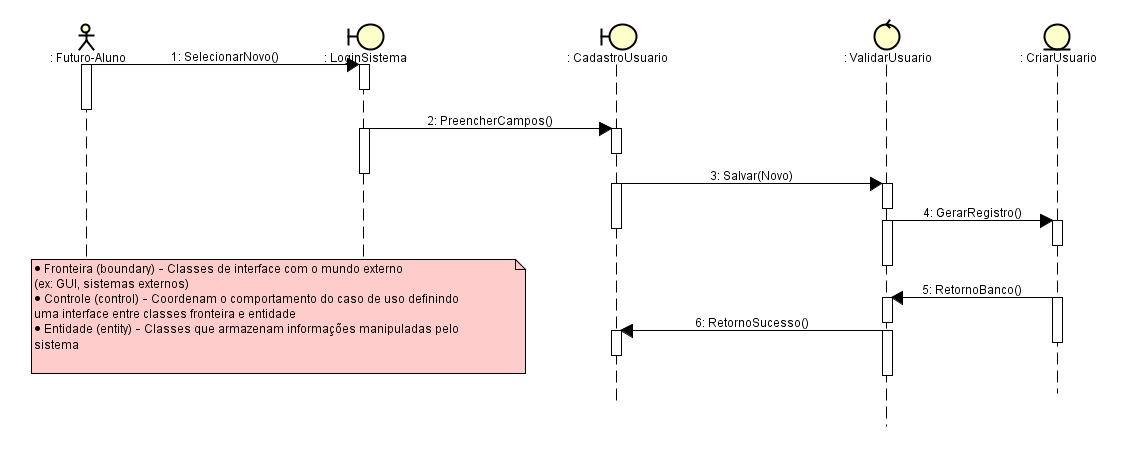
**Mensagens de retorno de validação:**

Mensagem devolvida pelas validações de regras de negócio.

**Operações do sistema:**

Para cada evento, há uma operação correspondente que o sistema desempenha. Podemos enxergar o sistema como possuindo operações, necessárias para descrever um comportamento dinâmico

*Figura 02 – Comportamento padrão de cenários.*



O diagrama de sequência sempre vai respeitar a arquitetura MVC. No caso o usuário via interface de uso aciona uma ação, assim acessando o controlador que valida as regras de negócio e faz acesso as entidades de banco. Obtendo a resposta do banco. A mensagem retorna para a tela. Ente telas muitas vezes não se faz necessário acessar entidades por exemplo.