

# COM220

## Computação

### Orientada a Objetos I

Aula 14: Introdução ao Modelo MVC

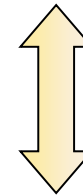
# O que é?

- O padrão MVC é um paradigma baseado em eventos que divide um sistema em 3 partes:
  - Modelo (Model)
    - Modelo de **dados**. Encapsula o estado do sistema
  - Visão (View)
    - Interface do usuário. Provê uma representação **visual** do modelo
  - Controle (Controller)
    - **Controle lógico** do sistema. Mapeia ações realizadas sobre uma dada visão em mudanças no modelo

# Arquitetura MVC

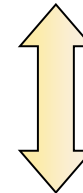


View



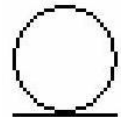
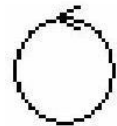
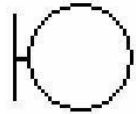
Inclui  
Informações

Controller



Altera o  
modelo

Model



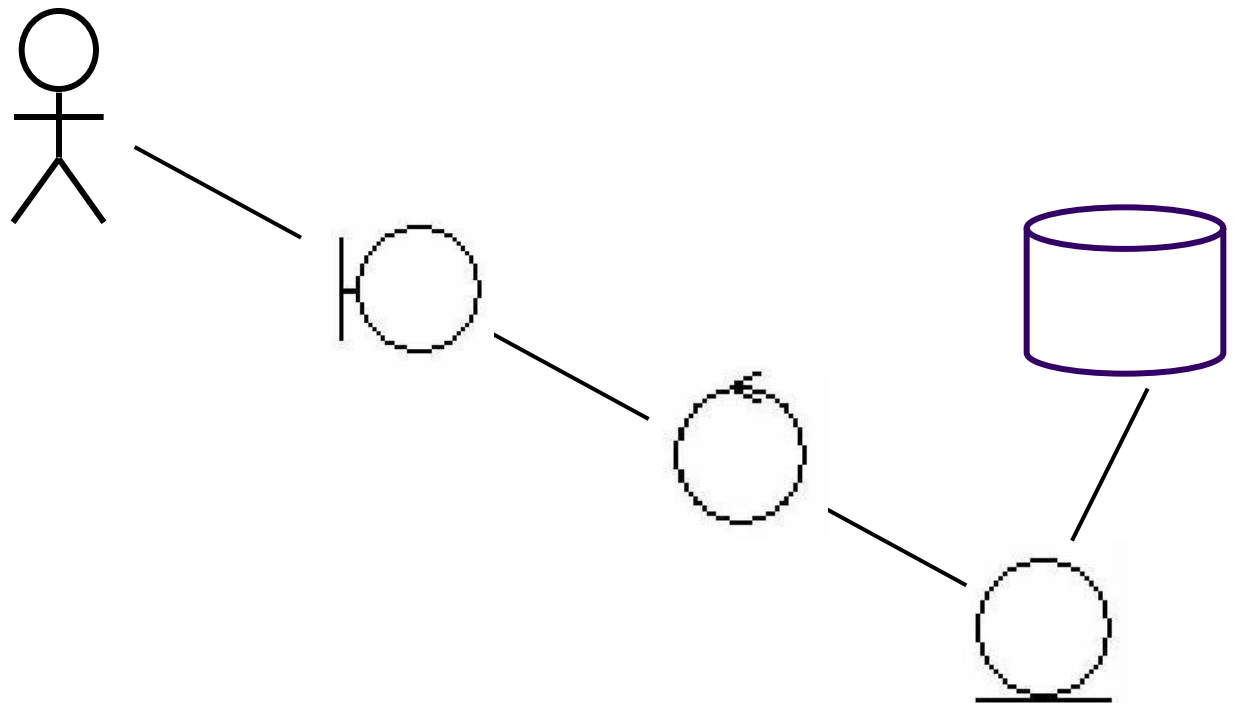
# Arquitetura MVC

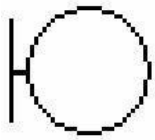
- Lida com a **separação de preocupações** (separation of concerns)
- Separa
  - **dados** e/ou lógica de negócio
    - da
  - **interface** do usuário
    - do
  - **controle** da aplicação
- Importante
  - O modelo do negócio não deve saber nada sobre as telas que exibem seu estado

# Arquitetura MVC

- A separação de preocupações é alcançada por meio de uma arquitetura a qual propõe a utilização de 3 tipos de classes

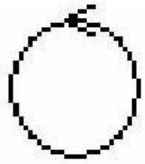
- Limite
- Controle
- Entidade





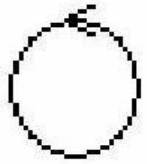
# Classe limite (view)

- Representa as **relações** entre os **atores** (mundo externo) e o **sistema**
- Seus objetos traduzem os eventos gerados por um ator em eventos relevantes ao sistema
- Normalmente tem as seguintes responsabilidades
  - Notificar os objetos de controle sobre eventos gerados externamente ao sistema
  - Notificar os atores sobre o resultado de interações entre objetos internos



# Classe controle (controller)

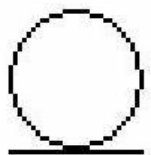
- É responsável pela **separação** entre as classes de interface (limite) e as classes do modelo
- Responsável por **controlar a lógica de execução** correspondente a um caso de uso
- Decide o que o sistema deve fazer quando um evento externo ocorre
  - Traduz eventos externos em operações que devem ser realizadas pelos demais objetos



# Classe controle (controller)

- Suas instâncias têm, normalmente, **vida curta**
  - Existem somente durante a realização de um caso de uso
- Responsabilidades típicas
  - Realizar monitorações, a fim de responder a eventos externos ao sistema (gerados por objetos limite)
  - **Coordenar a realização de um caso de uso**, enviando mensagens a objetos limite e objetos entidade
  - Assegurar que as regras de negócio estão sendo cumpridas



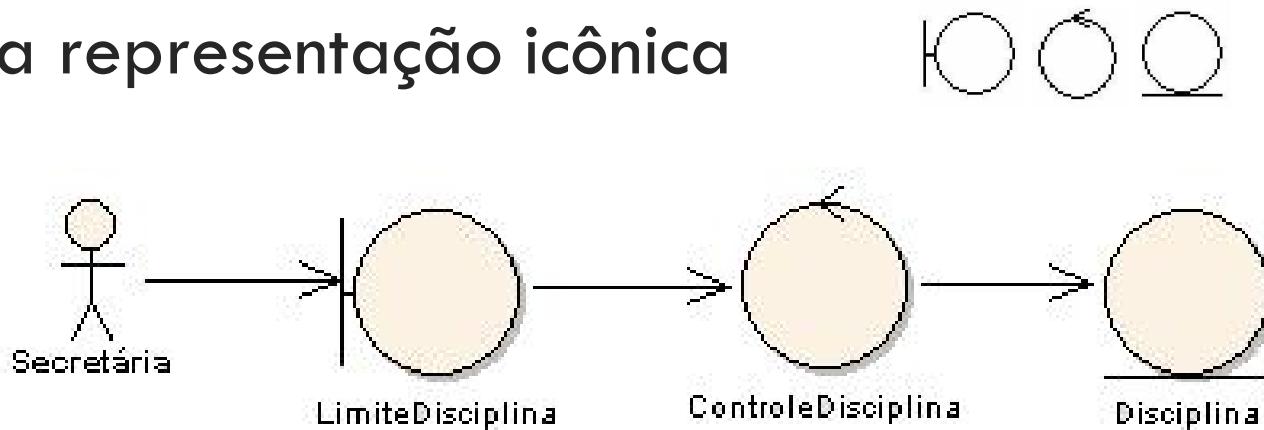


# Classe entidade (Model)

- São as **classes do domínio** do problema
  - Em um sistema acadêmico: curso, aluno, professor, ...
- Manipulam informação especializada e encapsulam o conhecimento do negócio
- Na maioria das vezes, manipulam **informações persistentes**. Por isso, podem ser usadas para gerar diretamente o esquema da base de dados
- **Atores nunca têm acesso direto a esse tipo de classe**
- Participam de vários casos de uso e têm ciclo de vida longo

# Análise de robustez

- Proposta por Ivar Jacobson, a partir do padrão MVC
- Fornece uma **visão de alto nível** de um sistema, na qual é possível identificar as classes limite, de controle e de entidade
- Utiliza representação icônica
- Ex:



# Exercício 1

11

- ❑ Implemente uma aplicação utilizando o modelo MVC para fazer o cadastramento de clientes, com nome e email
- ❑ A classe controladora deve manter uma lista com os clientes cadastrados

# Exercício 2

12

- ❑ Inclua um botão na sua implementação para listar os clientes cadastrados
  - ❑ Quando este botão for clicado, deve-se abrir uma nova janela e listar todos os clientes que constam na lista do controlador