

Desenvolvimento de Sistemas Orientados a Objetos I

Estruturando o Sistema Orientado a Objetos

Jean Carlo Rossa Hauck, Dr.

jean.hauck@ufsc.br

<http://www.inf.ufsc.br/~jeanhauck>

Conteúdo Programático

- Conceitos e mecanismos da programação orientada a objetos
- Práticas de Desenvolvimento de Software
 - Arquitetura em camadas e padrões de projeto

Estruturando um Sistema OO

- Um sistema Orientado a Objetos pode ser muito grande: milhares de classes, milhões de linhas de código
- É necessário definir uma estrutura que seja capaz de suportar o crescimento de um sistema OO desde o seu começo
- Dividir o sistema em camadas, de acordo com a responsabilidade

Arquitetura em camadas



[<http://cleveronsacramento.files.wordpress.com/2011/10/arquitetura-camadas.png>]

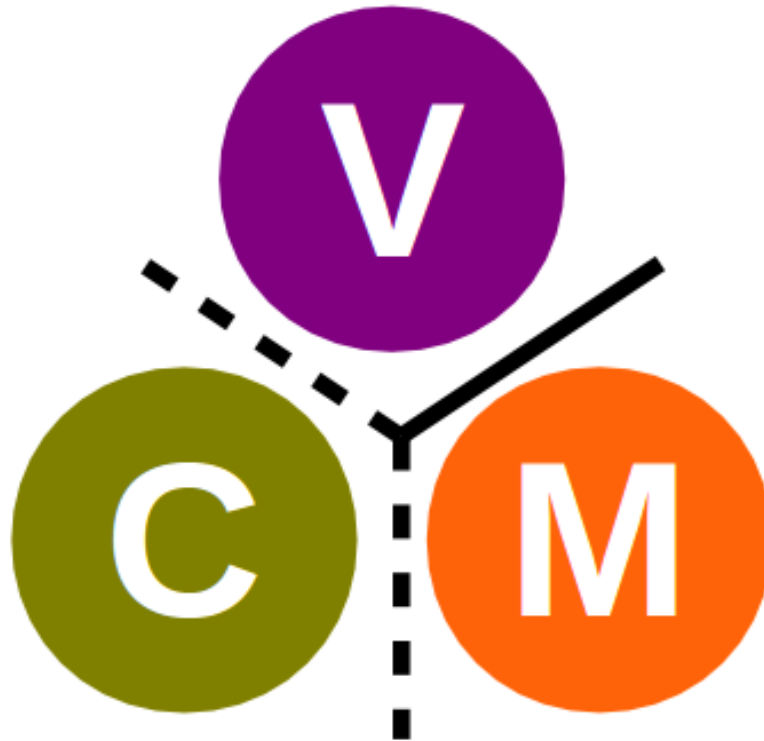
Arquitetura em camadas

- O objetivo é aumentar a coesão e reduzir o acoplamento
- Separação das classes por características assumidas dentro do contexto de uma aplicação
- Organização das classes em três camadas:
 - Limite, Controle e Entidade
 - ou, tipicamente:
 - Apresentação, Negócios e Persistência

Arquitetura em camadas

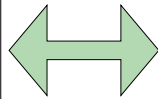
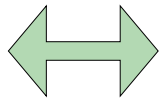
- **Limite (*boundary*)** → equivale ao **View** do padrão MVC
 - **Entrada e saída** de dados (usualmente telas, páginas)
 - Interação direta com os usuários do sistema
 - Também conhecidas como classes de fronteira
- **Entidade (*entity*)** → equivale ao **Model** do padrão MVC
 - Guarda informação em geral (dados)
 - Representam, usualmente, objetos **persistentes**
- **Controle (*control*)** → equivale ao **Controller** do padrão MVC
 - **Controle de fluxo** em geral (controla limites, entidades e outros controladores)
 - **Processamento** (algoritmos), regras de negócio

Como as camadas conversam entre si?

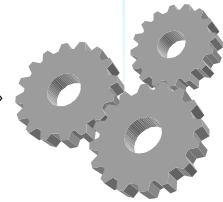
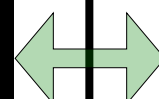


[<http://cleversonsacramento.files.wordpress.com/2011/10/mvc.png>]

View: Foco na entrada e saída de dados



```
class Conteudo:  
    def __init__(self, **kwargs):  
        self.dados = kwargs  
    ...
```



Usuário

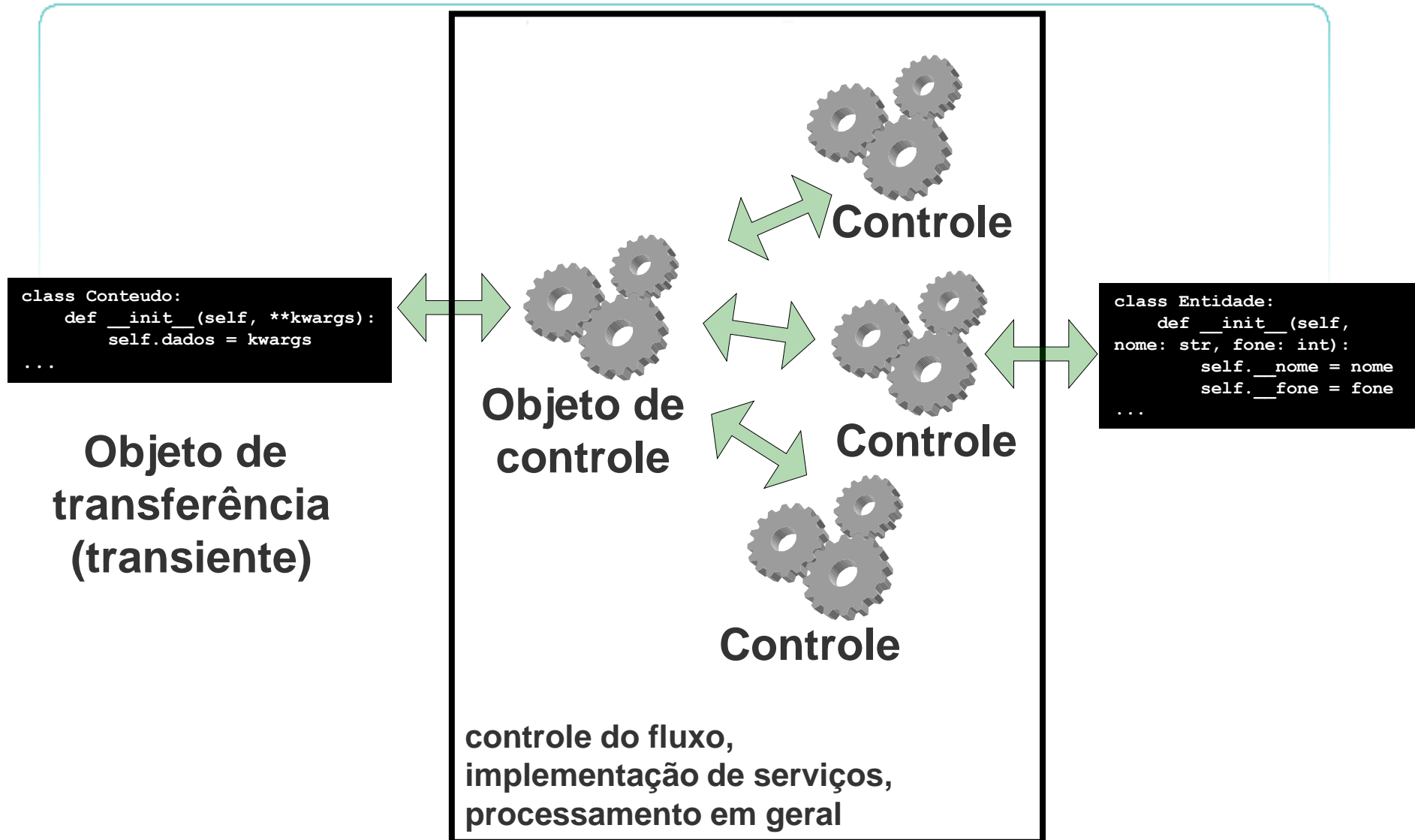
Objeto de
Limite

Objeto de
transferência
(transiente)

Objeto de
controle

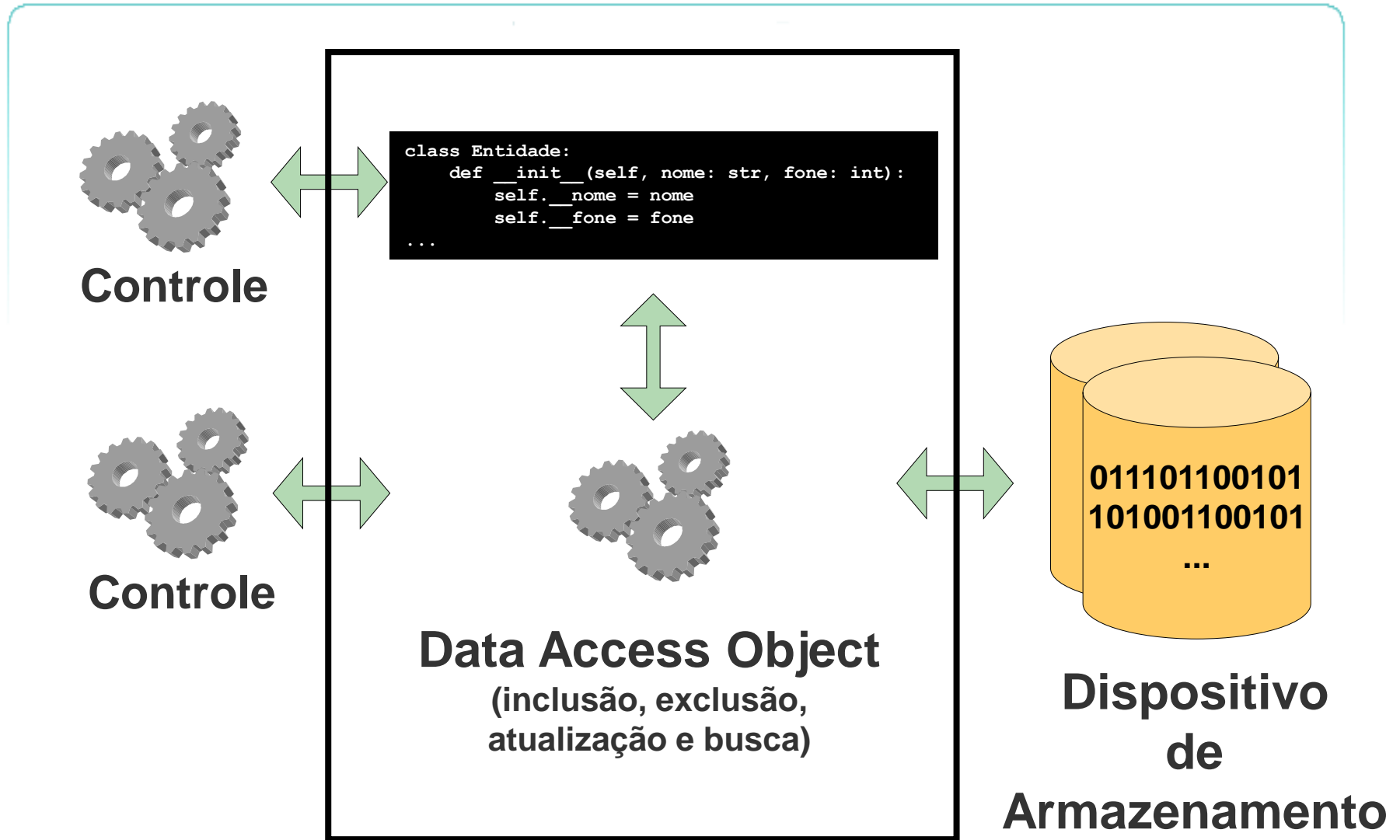
Baseado em [THIRY, 2006]

Controller: Foco no processamento



Baseado em [THIRY, 2006]

Model: Foco nos dados



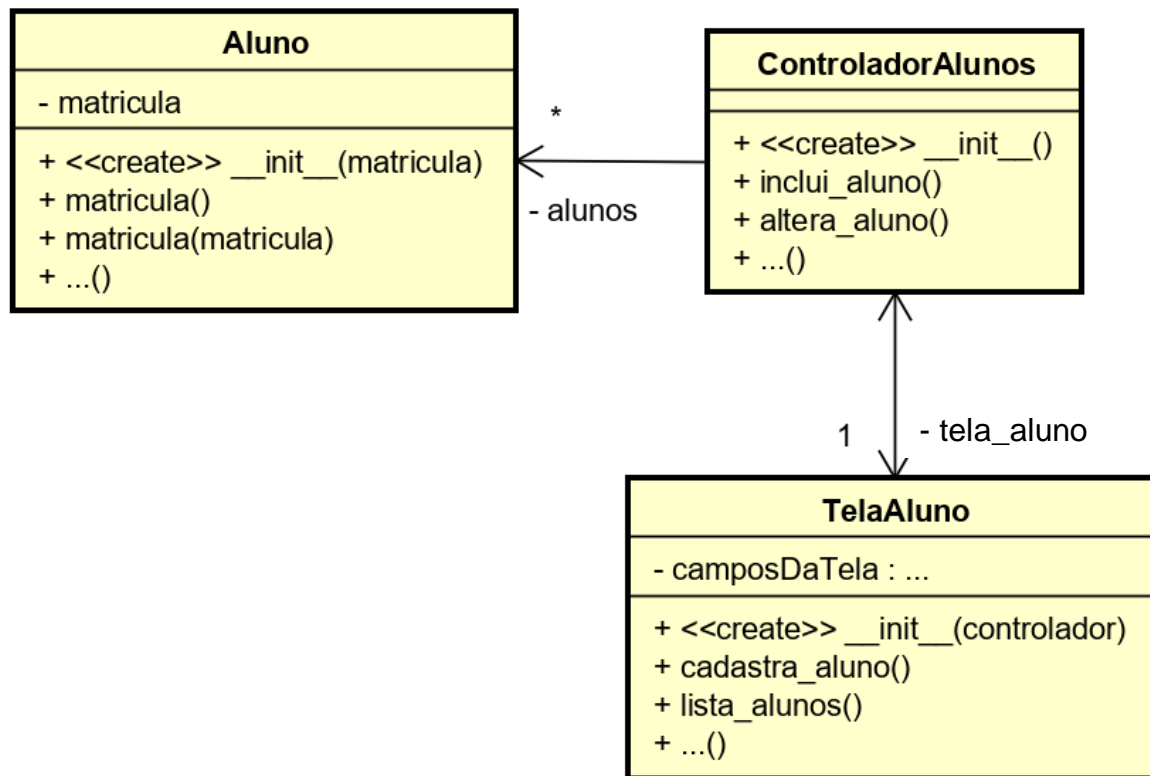
Baseado em [THIRY, 2006]

Então ... na prática

- Para cada Entidade, tipicamente mais duas classes:
 - **Controlador**: fluxos entre telas, regras de negócio que se apliquem a mais de uma instância, proprietário da tela, contato com outros controladores.
 - **Tela**: interação com o usuário, tratamento de conversões de tipos de dados, validações básicas.

Então ... na prática

- Para cada Entidade, tipicamente mais duas classes:



Agradecimento

Agradecimento ao prof. Marcello Thiry pelo material cedido.





Atribuição-Uso-Não-Comercial-Compartilhamento pela Licença 2.5 Brasil

Você pode:

- copiar, distribuir, exhibir e executar a obra
- criar obras derivadas

Sob as seguintes condições:

Atribuição — Você deve dar crédito ao autor original, da forma especificada pelo autor ou licenciante.

Uso Não-Comercial — Você não pode utilizar esta obra com finalidades comerciais.

Compartilhamento pela mesma Licença — Se você alterar, transformar, ou criar outra obra com base nesta, você somente poderá distribuir a obra resultante sob uma licença idêntica a esta.

Para ver uma cópia desta licença, visite <http://creativecommons.org/licenses/by-nc-sa/2.5/br/> ou mande uma carta para Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.