Aula 11

Professores:

Cláudia Maria Lima Werner

Leonardo Gresta Paulino Murta

Frameworks

Conteúdo:

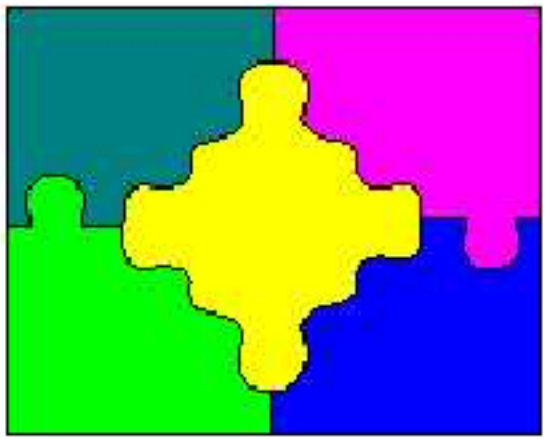
- *Frameworks*
- Projeto de *Frameworks*
- Construção de *Frameworks*
- Especialização de *Frameworks*
- *Framework* Caixa-Branca vs Caixa-Preta
- Dificuldades
- Bibliografia

Frameworks

Definição:

"Um Framework pode ser considerado como um projeto ou arquitetura de alto nível, consistindo de classes que são especialmente projetadas para serem refinadas e usadas em grupo."

[Wirfs-Brock e Johnson 1990]



Projeto de *Frameworks*

"O projeto de um framework é por si só uma atividade de pesquisa. O projetista deve entender as decisões de projeto e organizá-las em um conjunto de classes abstratas e concretas, juntamente com seus relacionamentos, que representam o comportamento básico para um subsistema. Logo, o projetista desenvolve uma teoria do domínio do problema e a expressa segundo um modelo orientado a objetos."

[Wirfs-Brock e Johnson 1990]

Projeto de *Frameworks*

"Um projeto de software é usualmente descrito em função de seus componentes e do modo como estes interagem. No framework, cada elemento principal do domínio do software é representado por uma classe (abstrata ou concreta), ou por um atributo, e a interação entre classes é feita através de mensagens."

[Johnson e Foote 1988]

Projeto de *Frameworks*

"O projeto de um framework não é uma tarefa tão objetiva quanto o desenvolvimento de uma aplicação específica. Algumas vezes, na busca da generalidade, são criados atributos, serviços e até mesmo classes que, em determinadas especializações ou instanciações do framework, podem não ser necessários.

O projeto de um framework requer muita experiência e experimentação, assim como o projeto das classes abstratas que o compõem."

[Johnson e Foote 1988]

Construção de *Frameworks*

- Regras para a construção [Johnson e Foote 1988]:
  - subdividir classes muito grandes;
  - encapsular separadamente elementos independentes do restante da classe;
  - separar métodos que não se comunicam;
  - reduzir a quantidade de passagens implícitas de parâmetros;
  - reduzir acoplamento entre os objetos, aumentando a sua coesão e generalidade.

Especialização de *Frameworks*

Segundo Wirfs-Brock e Johnson, o processo de especialização de um *framework* pode ser realizado através de duas formas básicas:

- Definição ou especificação de classes e subclasses necessárias e/ou;
- Configuração do conjunto de objetos existentes, através do fornecimento de parâmetros a cada uma das conexões entre estes.

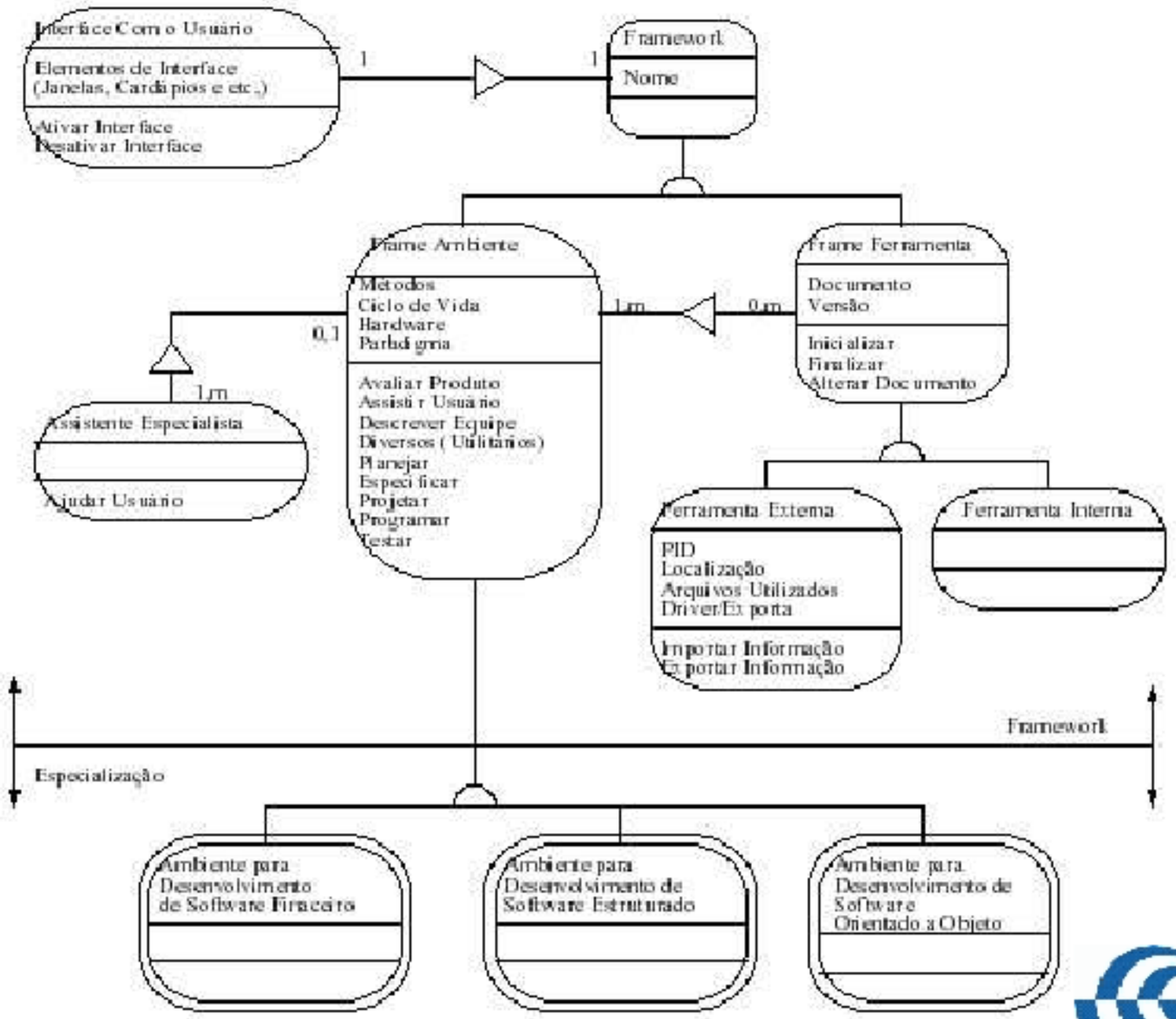
Framework Caixa-Branca

"Na primeira forma de especialização, o comportamento específico da aplicação é inserido na arquitetura genérica, somando-se métodos às subclasses de uma ou mais classes do framework. Cada método somado a uma subclasse deve estar de acordo com as convenções da respectiva superclasse.

Os frameworks que fazem uso deste processo de especialização obrigam quem os utiliza a conhecer os seus dispositivos internos, sendo portanto chamados de "frameworks caixa-branca."

[Johnson e Foote 1988]

Exemplo



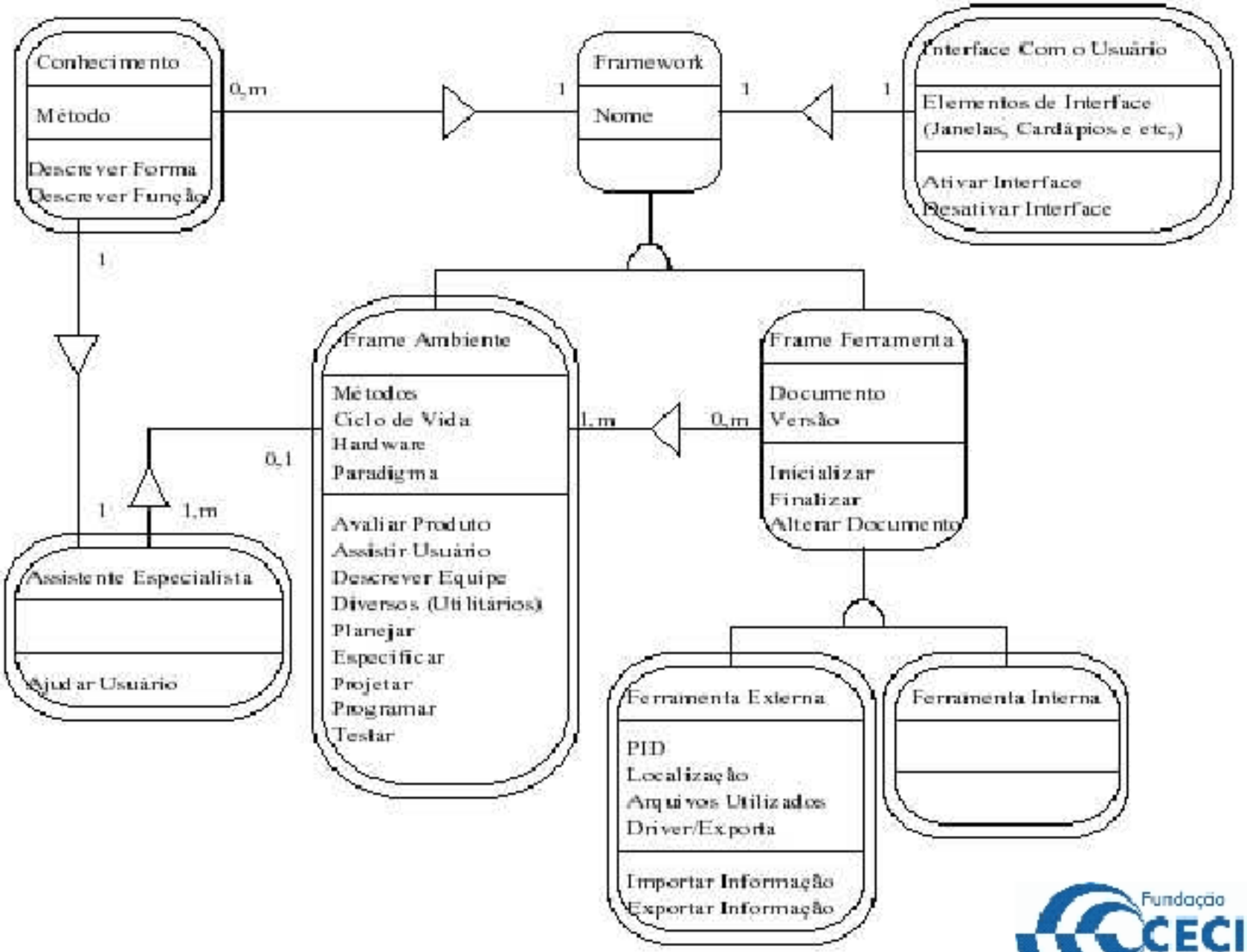
Framework Caixa-Preta

"Na segunda forma de especialização, o framework recebe um conjunto de parâmetros que representa o comportamento específico da aplicação. Este fornecimento de parâmetros é feito por protocolo, através da interface de cada classe do framework e, por isso, esta forma de especialização requer que seja conhecida apenas esta interface.

Consequentemente, o tipo de framework, assim especializado, passou a ser conhecido como "framework caixa-preta."

[Johnson e Foote 1988]

Exemplo



Especialização de *Frameworks*

- Vantagens de cada tipo de *framework*, quanto a reutilização:
  - O "*framework* caixa-preta" facilita a reutilização, pois exige que apenas a sua interface seja compreendida, podendo ser portanto encarado como um grande componente que encapsula o comportamento genérico de um domínio;
  - Os "*frameworks* caixa-branca" são mais flexíveis e permitem sua utilização na especificação de aplicações até então desconhecidas dentro do domínio.

Dificuldades

- Questões a serem abordadas:
  - dificuldade na compreensão sobre os objetivos, aplicabilidade e conseqüências do uso de *frameworks*;
  - garantia da qualidade de *frameworks*;
  - a construção de ambientes de suporte ao desenvolvimento e uso de *frameworks* é importante para garantir a eficiência e confiabilidade dessas atividades.

Bibliografia

- Surveying Current Research in Object-Oriented Design; R. Wirfs-Brock e R. Johnson; Communications of the ACM, 33 (9); setembro 1990
- Designing Reusable Classes, R. Johnson e B. Foote; Journal of Object-oriented Programming, 1 (2); jun/jul 1988
- Utilizando UML e Padrões - Terceira Edição, Larman, Bookman, 2007 (seção 38.3)