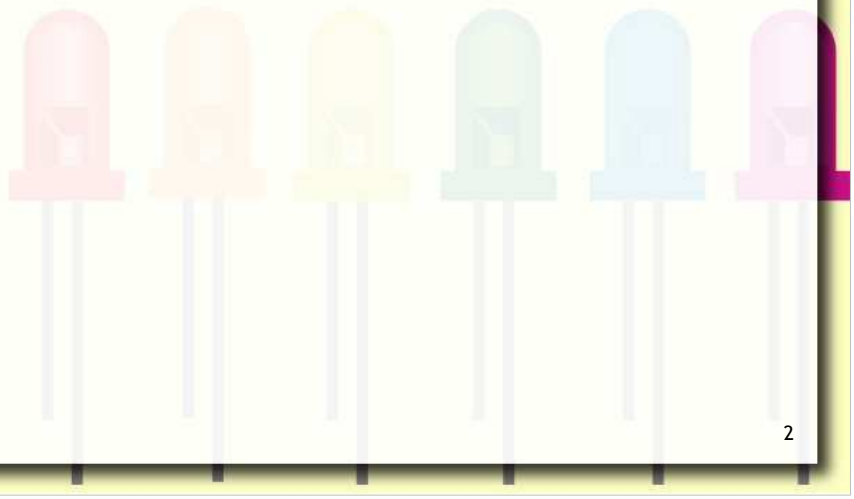
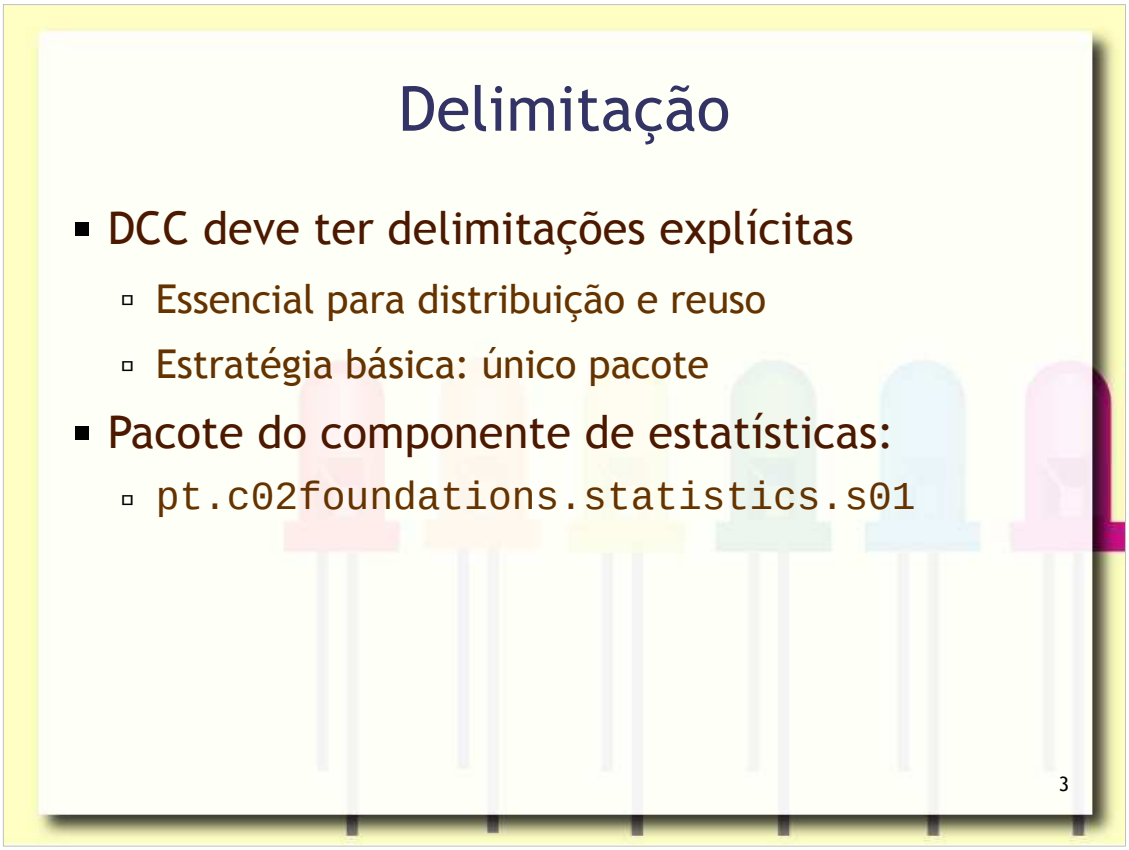
Minitutorial acompanha os slides nesta seção de comentários.

Objetivo do DCC

- Registrar um conjunto de números e calcular a soma e média destes números.



Todo o DCC deve ter um objetivo claramente definido. Este objetivo deve responder à seguinte pergunta: qual o serviço que este DCC presta?



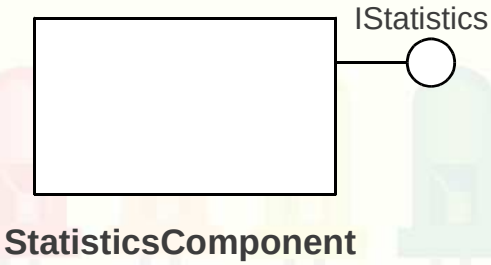
Cada DCC deve possuir claras delimitações, de modo que se possa diferenciar um componente do outro na etapa de distribuição e reuso.

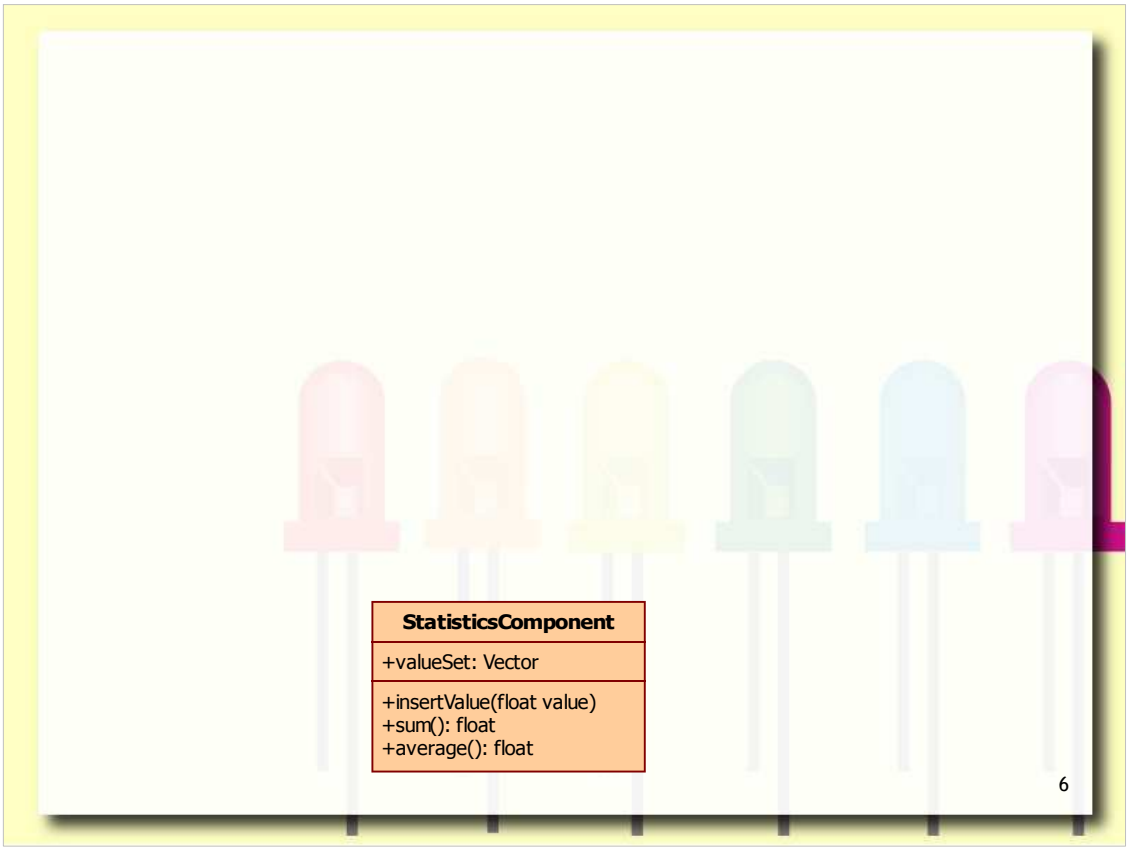
Delimitar significa estabelecer o que faz parte do componente e o que não faz.

A estratégia básica (mais simples) consiste em colocar o componente inteiro em um pacote, que é exclusivo para ele.

O ponto de partida é o projeto do DCC, que será mostrado a seguir em UML.

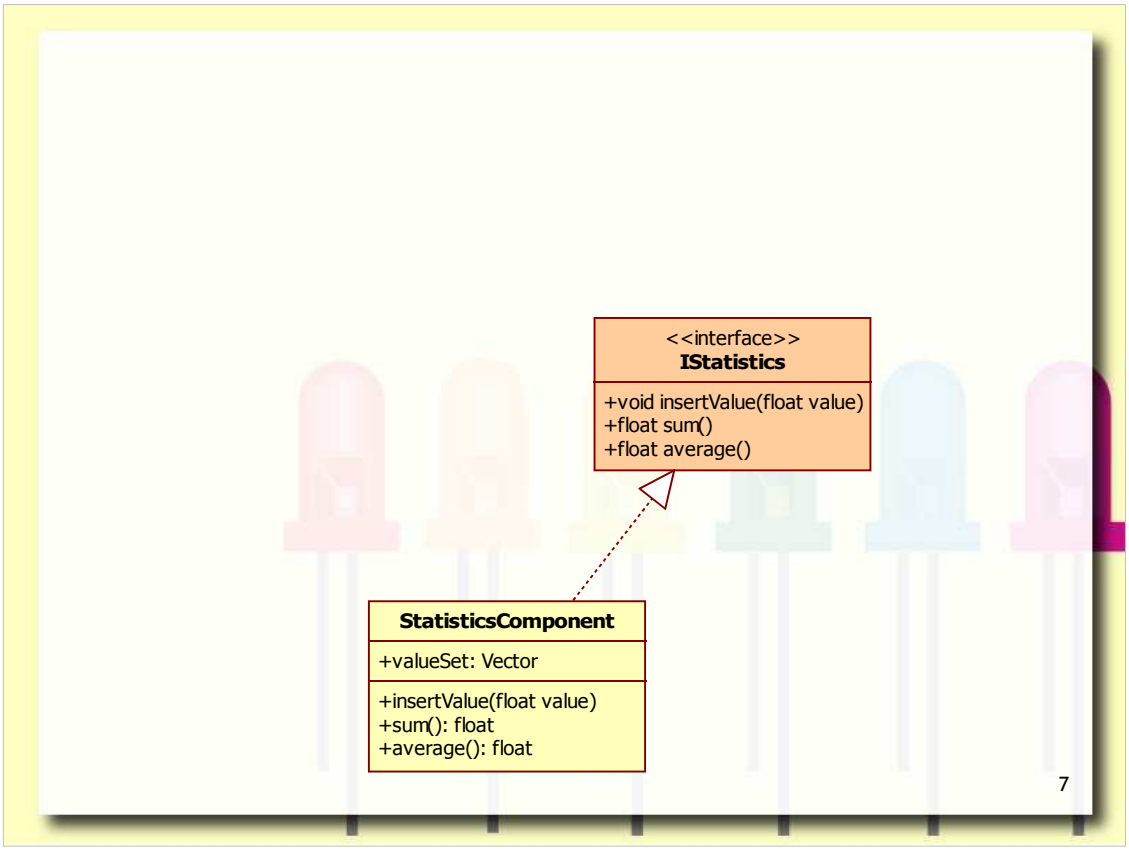
Componente e Interface Provida



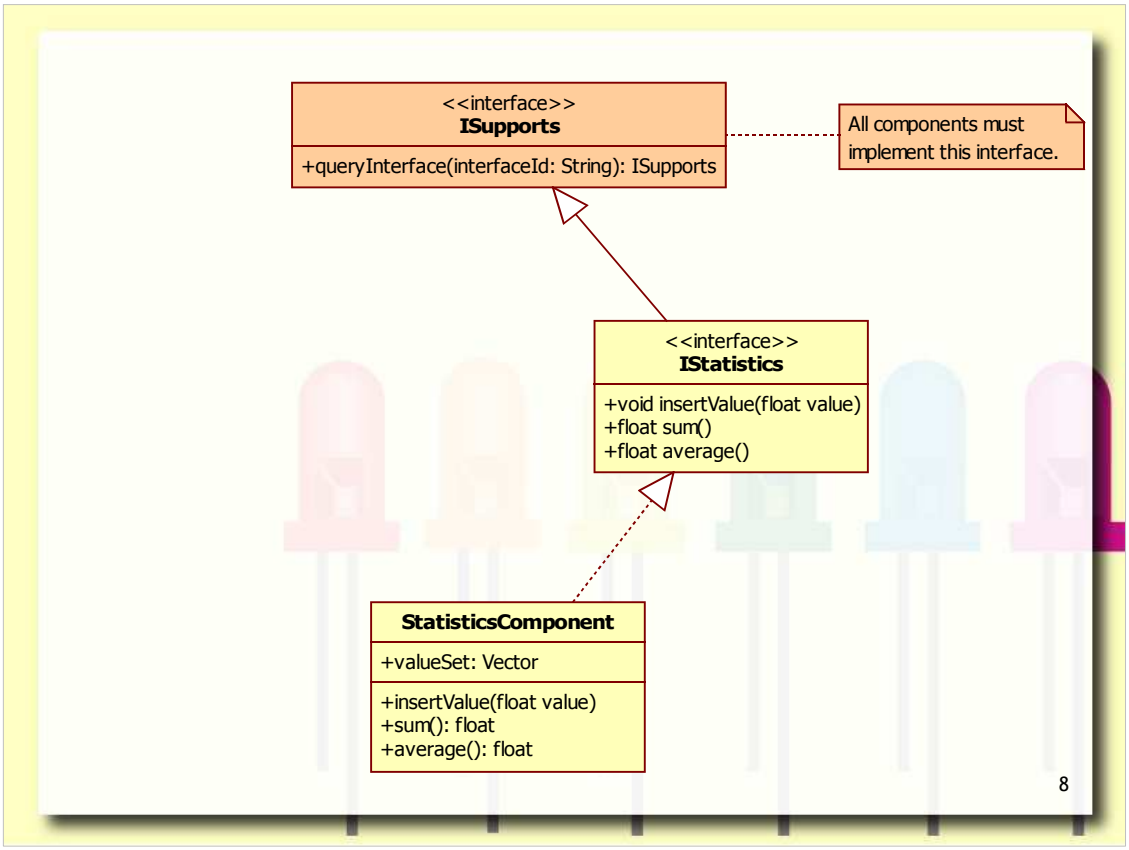


Cada DCC pode ser composto de uma ou mais classes. Neste tutorial será apresentado o caso mais simples em que cada DCC corresponde a uma classe Java.

Neste exemplo a classe StatisticsComponent implementa o DCC.

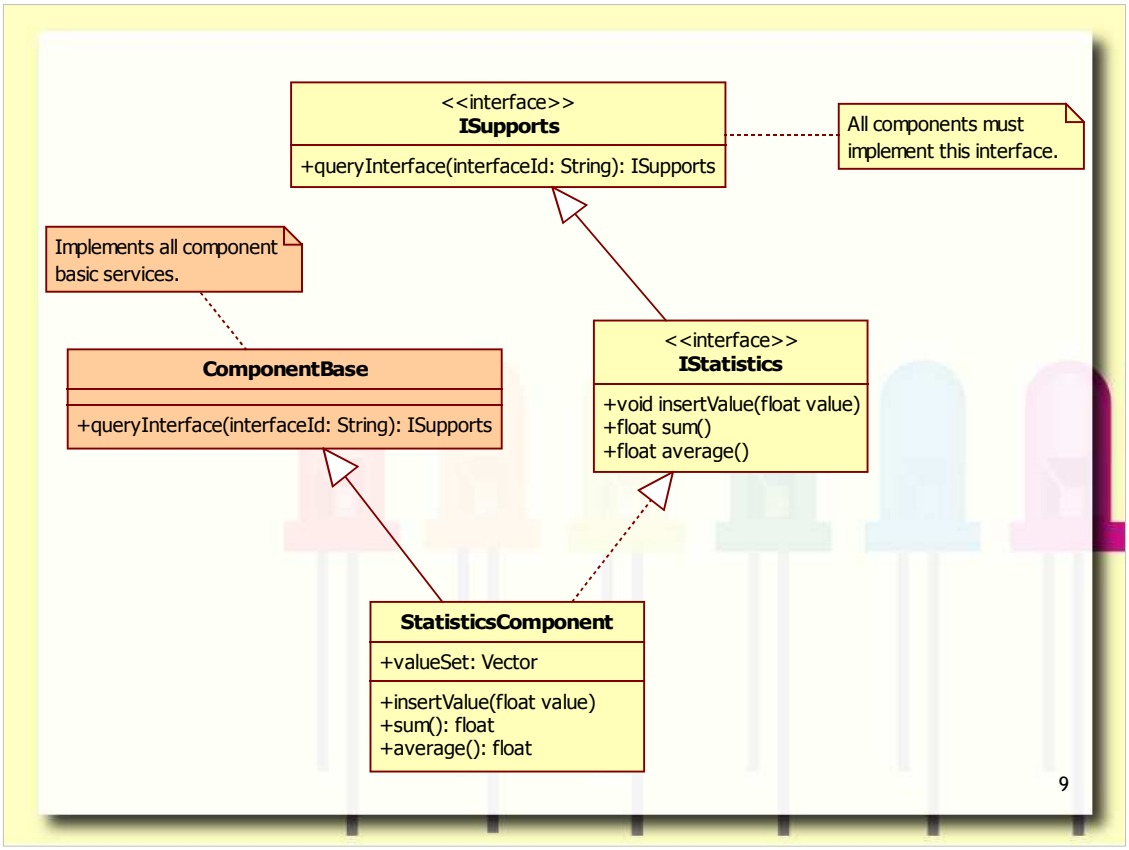


O DCC deve expor seus serviços na forma de uma ou mais interfaces, neste caso a interface IStatistics.

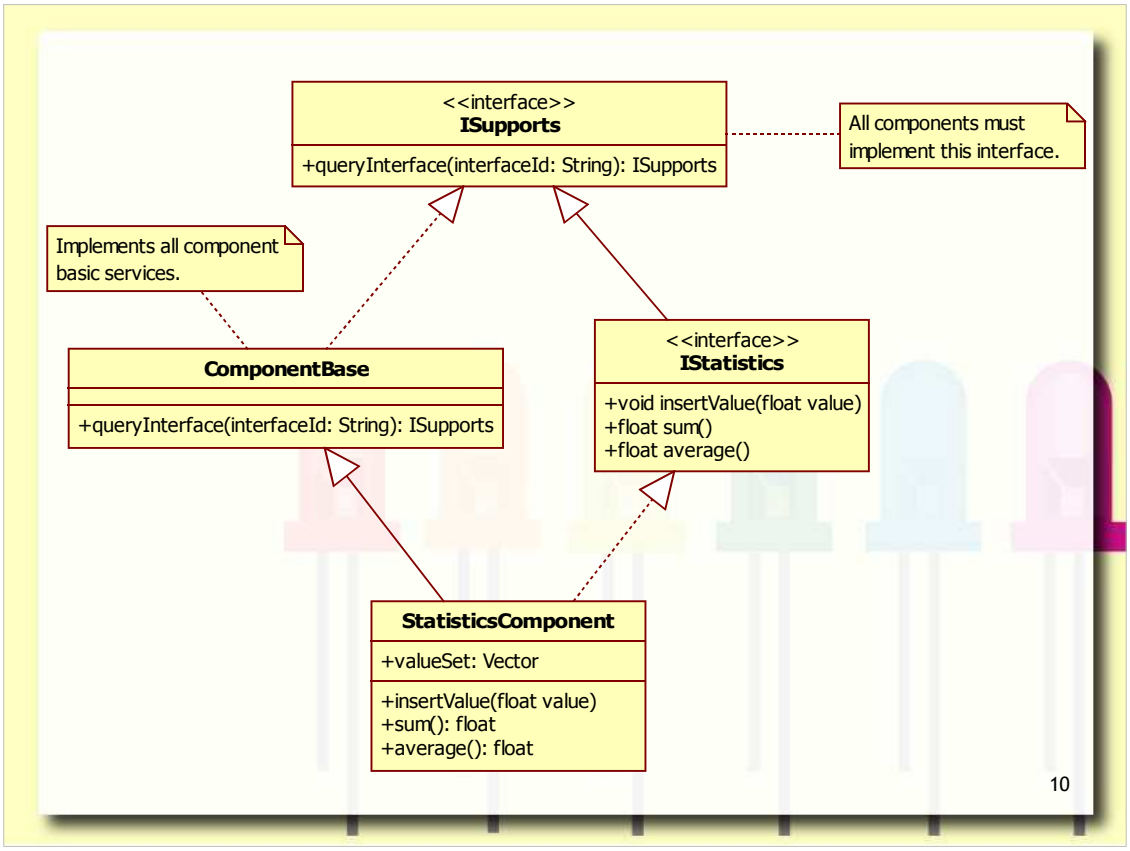


Todo DCC precisa implementar a interface ISupports. Além disto, todas as interfaces implementadas pelo DCC devem ser herdeiras de ISupports. Se o DCC implementa qualquer interface herdeira de ISupports ele já estará implementando ISupports.

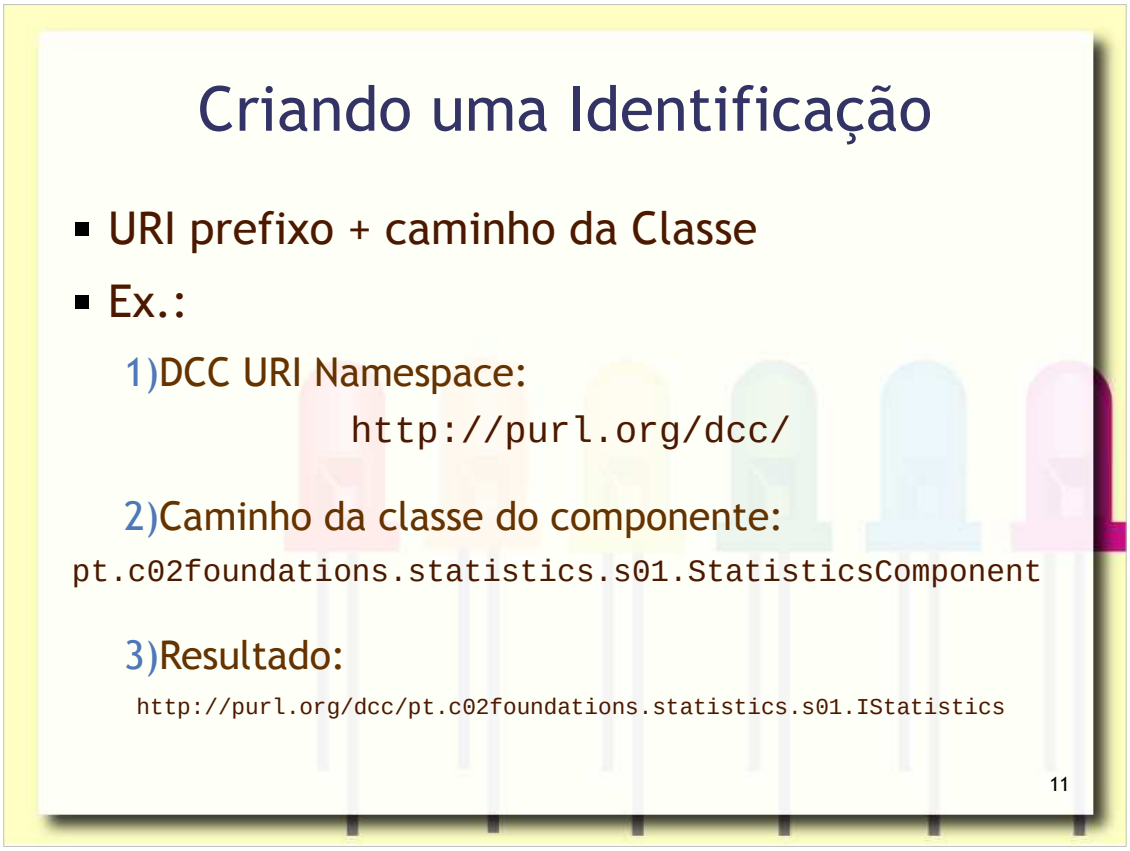
O interface ISupports define o conjunto de operações mínimas necessárias que qualquer DCC deve implementar.



Um DCC especial denominado ComponentBase implementa todas as operações de ISupports. Por esta razão, é usual (mas não obrigatório) que um DCC estenda ComponentBase.

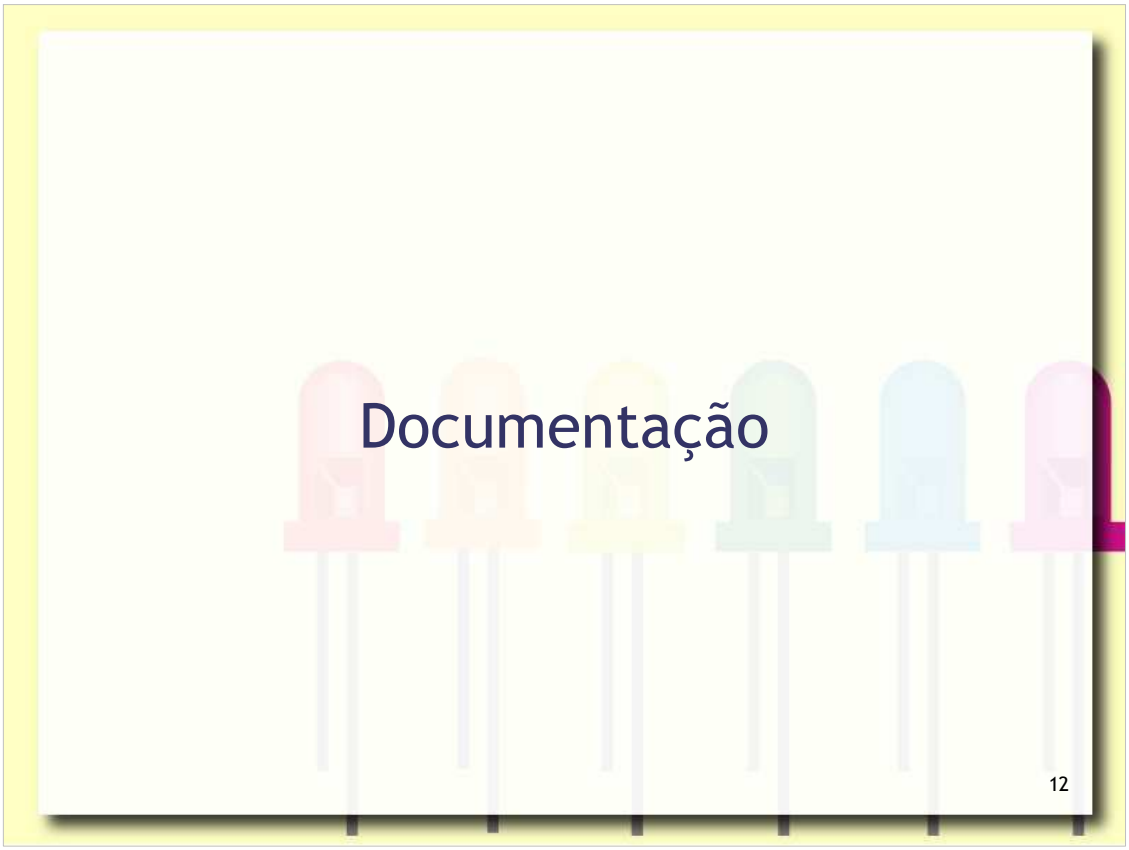


Deste modo, o StatisticsComponent não precisará reimplementar as operações exigidas por ISupports, dado que as herda de ComponentBase, mas pode estendê-las para acrescentar especificidades.



Os DCCs recebem um identificador que é único não apenas localmente, como também no contexto da Web, espaço onde eles serão compartilhados e reusados.

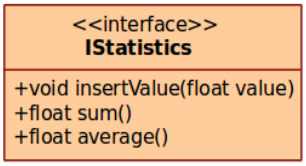
Para garantir que não haverá duplicidade, sugere-se a construção de identificadores baseados em URIs, compostos de duas partes: um prefixo de uma URI concatenado com o caminho da classe (incluindo o pacote).



Antes mesmo de se iniciar a codificação, sugere-se que seja criada a documentação do DCC. Dois documentos são fundamentais neste processo: um que descreve a interface e outro que descreve a classe.

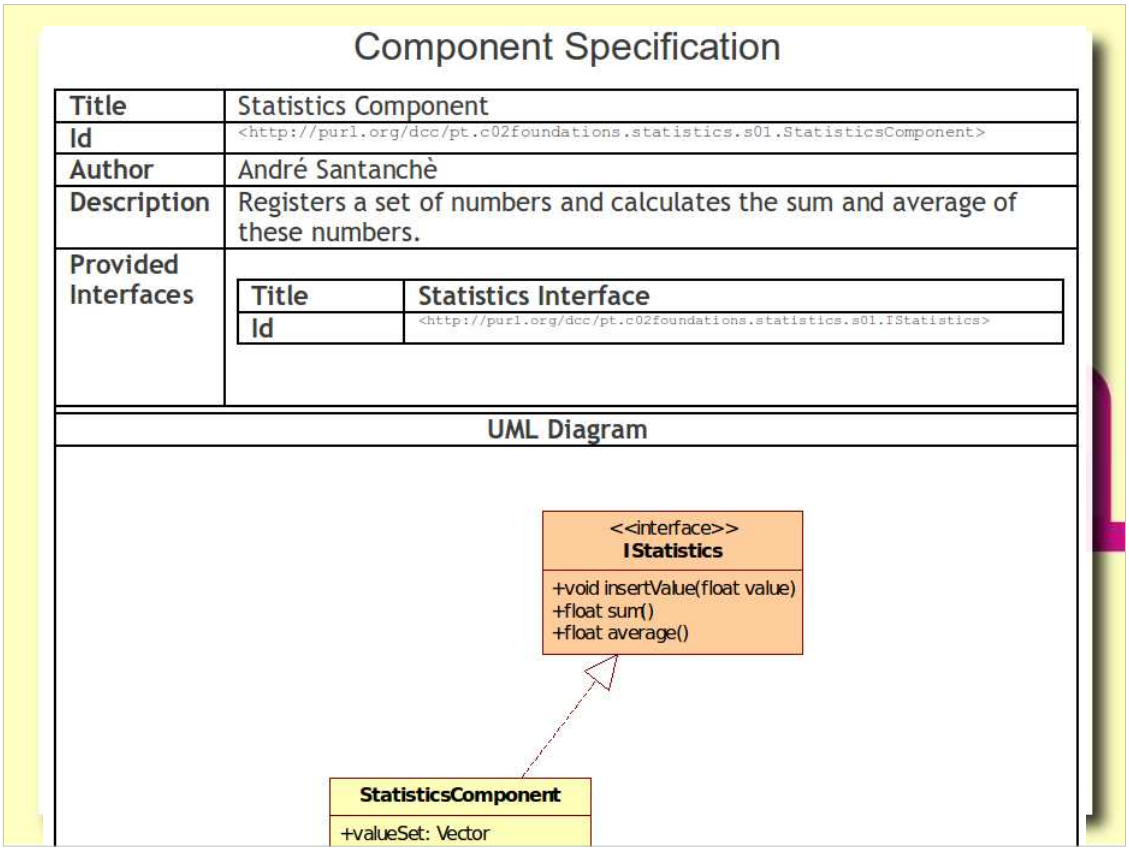
Interface Specification

Title	Statistics Interface	
Id	<http://purl.org/dcc/pt.c02foundations.statistics.s01.IStatistics>	
Author	André Santanchè	
Description	Interface for a Statistics Component that registers a set of numbers and calculates the sum and average of these numbers.	
Methods		
insertValue	Insert a value into the set.	
	value	the value to be inserted into the set
sum	Return the sum of the values in the set. Return zero if the set is empty.	
	return	sum of the values in the set
average	Return the average of the values in the set. Return zero if the set is empty.	
	return	average of the values in the set
UML Diagram		



A estrutura do documento para descrição da interface é mostrada acima. Ele requer:

- O título da interface
- O identificador único criado na etapa anterior
- Os autores
- Uma descrição sucinta do papel da interface
- Uma descrição dos métodos, incluindo a descrição dos seus parâmetros e retorno
- O diagrama UML da interface



A estrutura do documento para descrição do componente é mostrada acima. Ele requer:

- O título do componente

- O identificador único criado na etapa anterior

- Os autores

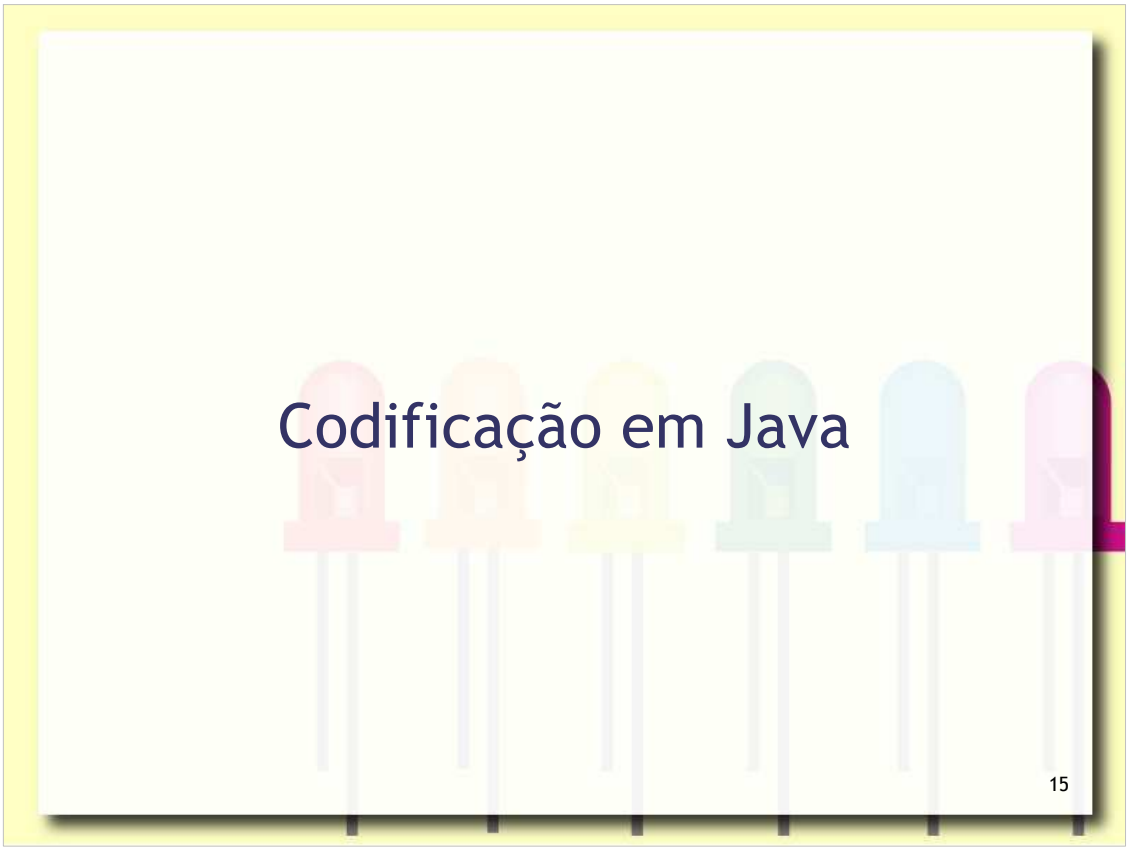
- Uma descrição sucinta do que faz o componente

- A listagem das interfaces providas e requeridas do componente

  . as interfaces providas são aquelas que definem serviços prestados; a interface criada anteriormente é provida

  . as interfaces requeridas são aquelas que definem serviços de outros componentes a ser usados pelo componente corrente

- O diagrama UML do componente



Os registros feitos na ficha se convertem em documentação do código de duas maneiras:

- seguindo o padrão Javadoc
- anotações @

como é mostrado nesta sequência de slides.

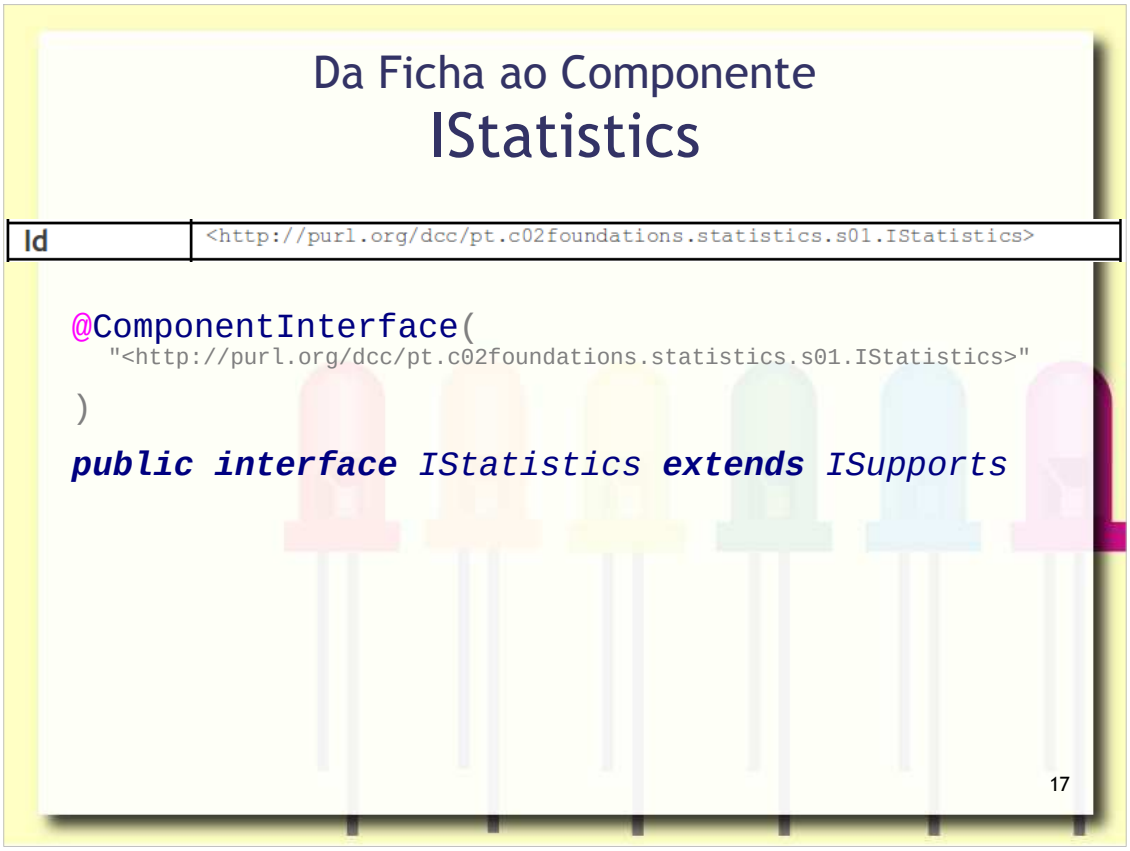
Da Ficha ao Componente

IStatistics

Author	André Santanchè
Description	Interface for a Statistics Component that registers a set of numbers and calculates the sum and average of these numbers.

```
/**
 * Interface for a Statistics Component that
   registers a set of numbers
 * and calculates the sum and average of these
   numbers.
 *
 * @author Andre Santanche
 */
public interface IStatistics extends ISupports
```


O autor e a descrição se convertem em comentários Javadoc.



O identificador URI da interface se converte em uma anotação @ComponentInterface.

Note que tanto na documentação como na anotação a URI fica entre <>.

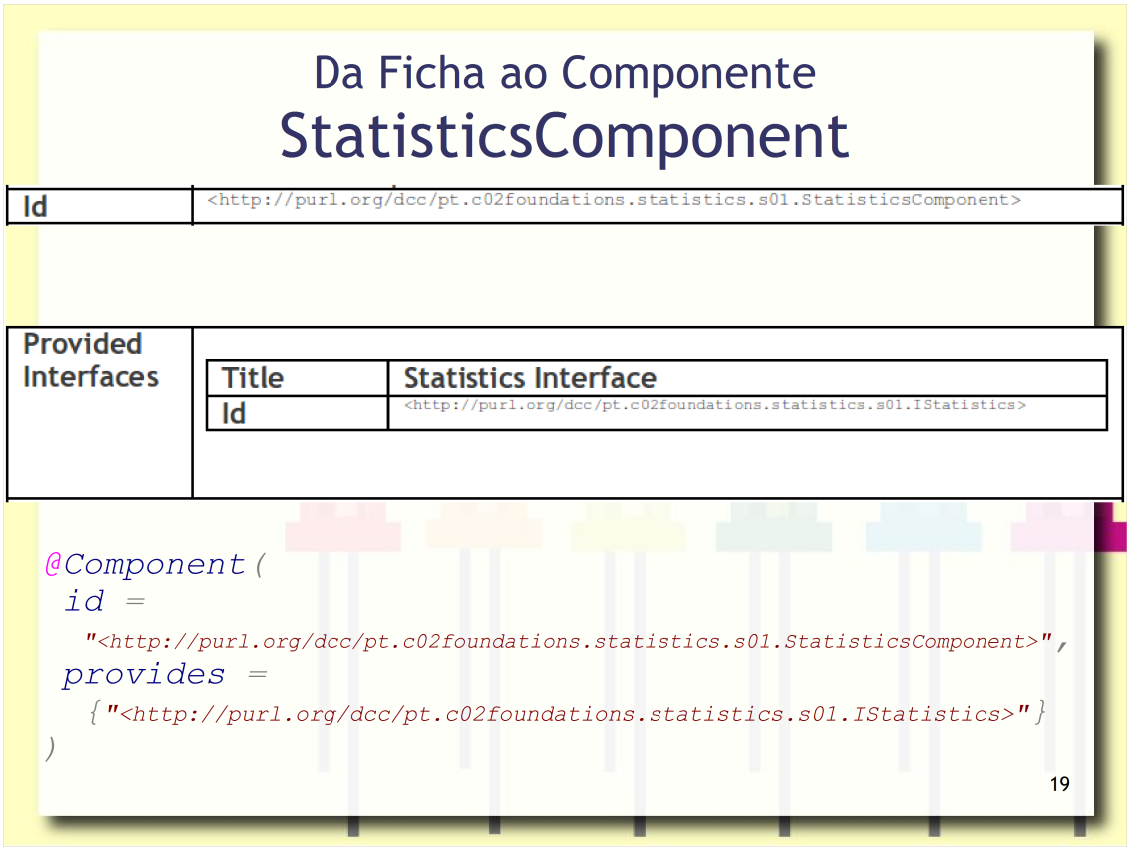
Da Ficha ao Componente

IStatistics

insertValue	Insert a value into the set.	
	value	the value to be inserted into the set

```
/**
 * Insert a value into the set.
 * @param value the value to be inserted into
   the set
 */
public void insertValue(float value);
```


A descrição dos métodos e dos parâmetros se convertem em Javadoc.

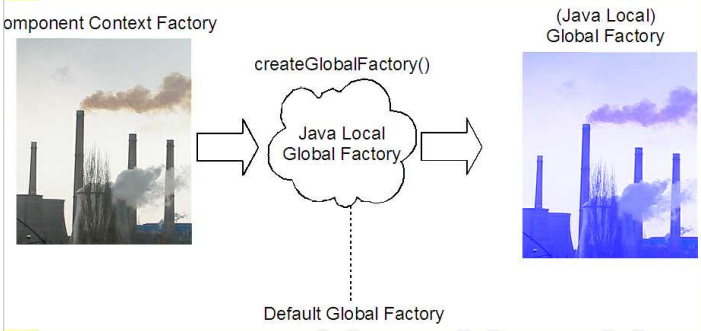


O identificador URI do componente, bem como a sua interface provida se convertem na anotação @Component.

A identificação deve ser identificada pelo campo id e a interface provida pelo campo provides.

Depois de montado o DCC é registrado na base e está pronto para uso, conforme será apresentado a seguir.

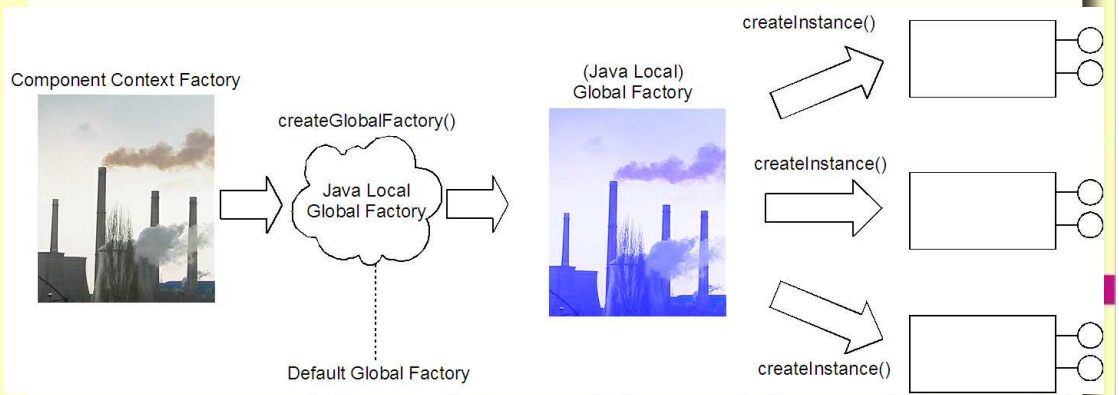
Criação da Fábrica Global



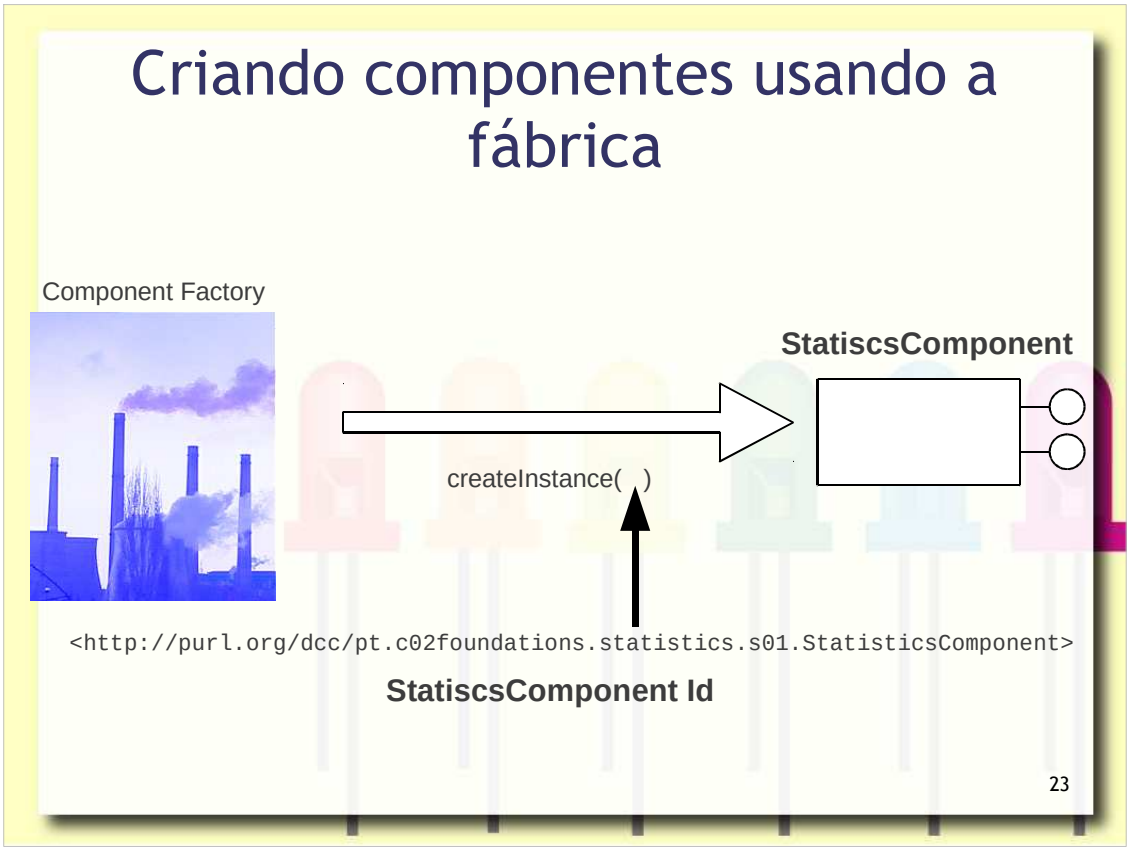
Os DCCs não devem ser criados pelo acionamento de seu construtor, através do comando new do Java, pois eles são montados por rotinas especiais chamadas fábricas.

Isto significa que antes do uso de qualquer DCC deve ser criada a sua fábrica. A forma mais simples de se fazer isto é através do método estático createGlobalFactory(), que pertence à classe ComponentContextFactory.

Criando componentes usando a fábrica



A mesma fábrica pode ser usada para a criação de tantos componentes quantos forem necessários.



O método createInstance da fábrica recebe como parâmetro o identificador URI do DCC e o cria. Como um DCC pode ter diversas interfaces, este método alternativamente possibilita a escolha da interface. Caso ela não seja escolhida, ele retornará a interface padrão. Quando o componente só tem uma interface (este é o caso), tal interface se torna a padrão.

Lembre-se de colocar os símbolos <> sempre que especificar a URI por extenso.

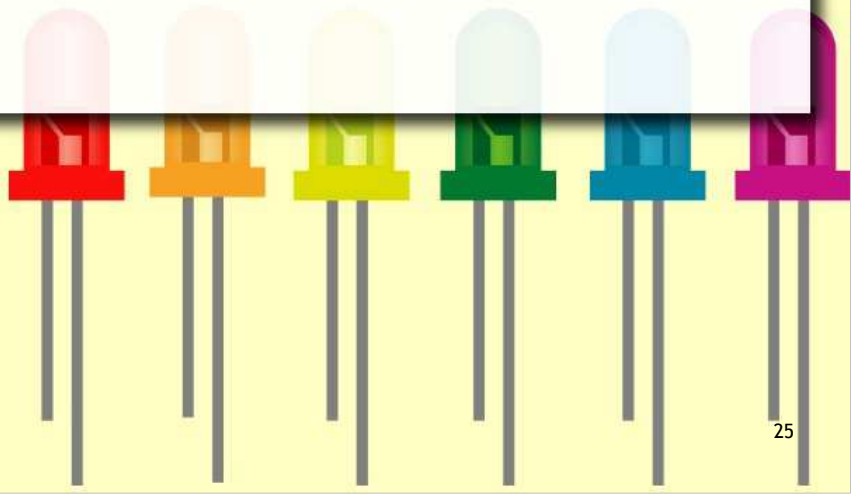
Referências

- Martin, R. C. **Design Principles and Design Patterns**. Object Mentor, 2000.
- Gamma, E. Helm, R. Johnson, R. Vlissides, J. **Design Patterns: Elements of Reusable Object-Oriented Software**. Addison-Wesley, 1995.

Algumas referências interessantes para começar a entender padrões de projeto que são usados neste framework, especialmente o padrão de fábricas.

André Santanchè

http://purl.org/andresantanche



License

- These slides are shared under a Creative Commons License. Under the following conditions: Attribution, Noncommercial and Share Alike.
- See further details about this Creative Commons license at: http://creativecommons.org/licenses/by-nc-sa/3.0/