# INSTITUTO FEDERAL DA PARAÍBA - IFPB CURSO SUPERIOR TECNOLÓGICO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

### ALEXA GONÇALVES LINS FRANCISCO RICARTE RODRIGUES SOARES THIAGO YURE VIEIRA DE ALMEIDA

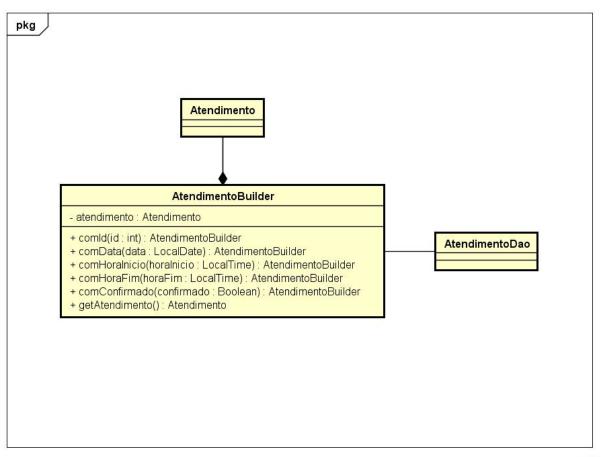
RELATÓRIO DO PROJETO FINAL DA DISCIPLINA DE PADRÕES DE PROJETO - SALON

CAJAZEIRAS - PB ABRIL DE 2018

## SUMÁRIO

	BUILDER	10
	PROTOTYPE	20
3.	FACTORY METHOD	30
4.	COMMAND	40

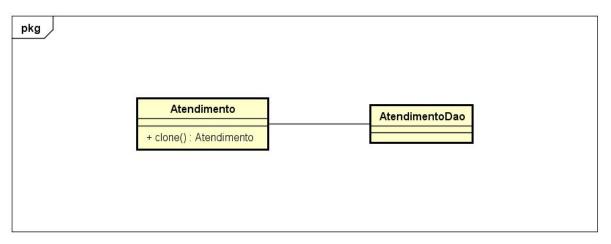
#### 1. BUILDER



powered by Astah

A utilização do padrão builder permitiu separar a construção da classe atendimento da sua representação, permitindo ter mais controle da sua construção, além de garantir maior legibilidade do código. Utilizando o padrão, foi possível utilizar a classe atendimento de modo a ser flexível para criar duas representações da mesma: uma referente a agenda do atendente, tendo apenas os atributos atendente serviço e ID com valores e outra referente ao atendimento em si, tendo todos os atributos da classe com valores.

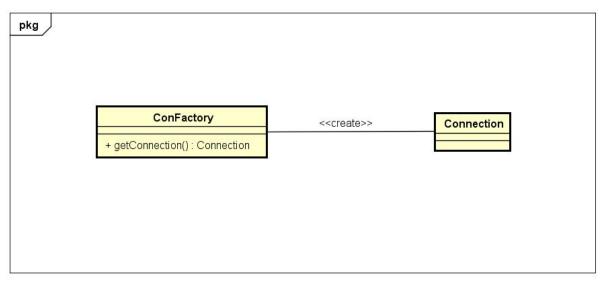
#### 2. PROTOTYPE



powered by Astah

Com a utilização do padrão, foi possível criar várias instâncias novas da classe atendimento derivadas de um objeto, também da classe atendimento, já que essa classe é um protótipo. Dessa forma é mais conveniente já que as novas instâncias da classe atendimento possui vários valores de atributos em comum com o objeto atendimento usado como protótipo, poupando reescrita desnecessária de código.

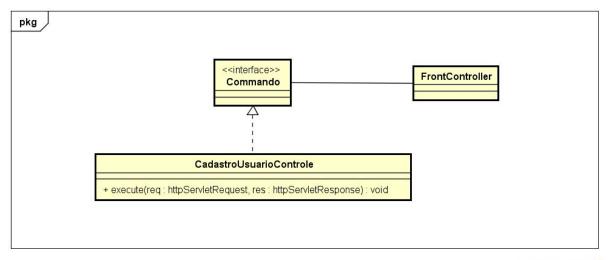
#### 3. FACTORY METHOD



powered by Astah

Através do uso do padrão foi possível encapsular a criação de uma conexão com o banco de dados, retirando essa responsabilidade das classes que necessitam de uma conexão, passando-as para a classe conFactory que possui o static factory method getConnection.

#### 4. COMMAND



powered by Astah

Utilizando o padrão foi possível flexibilizar o tratamento de requisições vindas das páginas JSP, criando objetos que encapsulam a solicitação, e implementando uma interface comum Command. Dessa forma, o uso de condicionais no FrontController foi reduzido.