

13

## Para saber mais: Métodos de callback

A anotação `@PostConstruct` sinaliza que aquele método deve se chamado logo após da criação e da injeção. Esse método também é chamado de **Callback**. O método faz parte do ciclo da vida do objeto, aqui quando o objeto nasce.

Como existe uma anotação que indica o início do ciclo, também existe um método que indica o fim do ciclo da vida!

Por exemplo, podemos colocar no `AutorBean` os seguintes métodos:

```
@Named
@ViewScoped //javax.faces.view.ViewScoped
public class AutorBean implements Serializable{

    //atributos omitidos

    @PostConstruct
    void init() {
        System.out.println("AutorBean está nascendo ....");
    }

    @PreDestroy
    void morte() {
        System.out.println("AutorBean está morrendo ....");
    }
}
```

Como o `AutorBean` está no escopo da tela ( `ViewScoped` ), ele vai morrer quando a tela morre. Isso pode ser um pouco difícil de simular. Para ver o nosso método `morte` realmente funcionar, é mais fácil colocar o `AutorBean` temporariamente no escopo da requisição ( `javax.enterprise.context.RequestScoped` ):

```
@Named
//@ViewScoped //javax.faces.view.ViewScoped
@RequestScoped //apenas para testar os callbacks
public class AutorBean implements Serializable{
```