



08

## Atributos de transação



46%

ATIVIDADES  
8 DE 9FÓRUM DO  
CURSOVOLTAR  
PARA  
DASHBOARD

### Será que tem problemas em trabalhar com transações de forma programática? Vamos fazer um teste!

1) Abra a classe `AuthService` e anote o método `adiciona()` com

```
@TransactionAttribute(TransactionAttributeType.REQUIRED) .
```

Depois, faça o método lançar uma *exception* após chamar o método `salva()` do DAO:

```
@TransactionAttribute(TransactionAttributeType.REQUIRED)
public void adiciona(Autor
autor) {
    this.dao.salva(autor);
    throw new
RuntimeException(
        "[ERRO] Erro
        lançado para testar o rollback
        da transação.");
}
```

COPIAR CÓDIGO

2) Dentro de `AutorDao`, modifique o método `salva()` para que ele suspenda a transação corrente, abra uma nova, realize o *commit*, encerre a nova transação e reative a transação que havia sido suspensa no início do processo.

Para isso, basta anotar o método com

```
@TransactionAttribute(TransactionAttributeType.REQUIRES_NEW)
```



116.5k xp





46%

ATIVIDADES  
8 DE 9FÓRUM DO  
CURSOVOLTAR  
PARA  
DASHBOARD

```
@TransactionAttribute(TransactionA1
    public void salva(Autor
autor) {
    ...
}
```

COPIAR CÓDIGO

3) Faça o *Full Publish* da aplicação no JBoss AS e depois limpe o console do Eclipse para ver melhor as mensagens de saída da nossa aplicação.

4) Vamos testar o cadastro de Autor. Tente incluir um novo autor! Ao clicar em salvar parece que nada acontecer, certo? Olhe o console do Eclipse. Foi lançada a *exception* que criamos no `AutorService`, sendo assim, a transação foi encerrada com erro, um *rollback* foi realizado e nada foi gravado no banco, certo?

5) Abra o terminal de comandos do MySQL e consulte todos os elementos da tabela `Autor` no banco `Livraria`:

```
$ mysql -u root
mysql> use Livraria;
mysql> select * from Autor;
```

COPIAR CÓDIGO

6) E aí? Mesmo a aplicação tendo apresentado um erro no console e realizado *rollback* na transação, por que você acha que o autor foi inserido no banco assim mesmo?



116.5k xp





46%

ATIVIDADES  
8 DE 9FÓRUM DO  
CURSOVOLTAR  
PARA  
DASHBOARD

## Opinião do instrutor



A resposta desta questão está totalmente relacionada aos **Atributos de Transação**.

Quando usamos o atributo

```
@TransactionAttribute(TransactionAttributeType.REQUIRED)
```

no ***service***, dissemos ao *container* que a operação deve obrigatoriamente ser executada dentro de uma transação. Mas quando colocamos o atributo

```
@TransactionAttribute(TransactionAttributeType.REQUIRES_NEW)
```

no método `salva()` do DAO, o método ao ser chamado, suspende automaticamente a transação corrente que veio do ***service*** e abre uma nova; que, por sua vez, será usada para salvar o autor no banco e em seguida será encerrada automaticamente pelo método, sem qualquer tipo de erro ou *exception* que force um *rollback*.

Quando a transação que havia sido suspensa pelo DAO é reativada, nossa *exception* é lançada e a transação é forçada a fazer um *rollback*. Mas esse *rollback* não interferiu no cadastro do autor no banco porque pela escolha dos **atributos de transação**, as transações eram diferentes e independentes entre si.



116.5k xp

