

Mãos à obra: Criação do interceptor

Com a anotação criada, podemos criar o interceptor com CDI! Vamos tentar?

1) No pacote `br.com.caelum.livraria.tx` crie uma nova classe com o nome `GerenciadorDeTransacao` :

```
public class GerenciadorDeTransacao implements Serializable {  
  
}
```

2) A classe também vai utilizar a anotação `@Transacional` , além da `@Interceptor` :

```
@Transacional  
@Interceptor  
public class GerenciadorDeTransacao implements Serializable {
```

3) Para podermos gerenciar a transação injete o `EntityManager` :

```
@Inject  
EntityManager manager;
```

4) Agora vamos implementar o método que realmente chama `begin` e `commit` :

```
@AroundInvoke  
public Object executaTX(InvocationContext contexto) throws Exception {  
  
    manager.getTransaction().begin();  
  
    // chamar os daos que precisam de um TX  
    Object resultado = contexto.proceed();  
  
    manager.getTransaction().commit();  
  
    return resultado;  
}
```

5) Por fim, declare o interceptor no `beans.xml` . Cuidado para não errar o nome do pacote:

```
<beans>  
    <interceptors>  
        <class>br.com.caelum.livraria.tx.GerenciadorDeTransacao</class>  
    </interceptors>  
</beans>
```

6) Agora, abra o o DAO genérico e apague todos as linhas que abrem (`begin`) e fecham (`commit`) a transação. Não deixe nenhum código no DAO que usar a transação.

7) Agora, com muita coragem, inicie o Tomcat e tente cadastrar um autor ou livro. Verifique sempre o console para pegar possíveis erros.