

# Peer instruction - Contexto de Persistência

## Atividades

- Leia o material indicado adiante. Os objetivos de aprendizagem são fornecidos adiante também. É o que se espera que você saiba fazer depois do estudo.
- Responda aos exercícios chamados "**Preparação para teste de leitura**". Eles não serão corrigidos mas servirão de preparação para o teste de leitura feito em aula.
- Fora da sala de aula, forme grupos alunos (sugestão de 2~3 alunos) e prepare uma resposta para o "**Teste de objetivos de aprendizagem**".
- Prazo para entrega: **24 de abril de 2017**, às 7hs.

## Material para leitura antes da aula

- [Material da aula](#) ou [Material](#)
- Capítulo 5 do livro **Enterprise Javabeans 3.0**. [Burke, B. 2007]
- Capítulo 5 do livro **EJB3 Profissional: Java Persistence API**. [Schincariol, M. 2008]
- Capítulo 9 do livro **EJB3 Em Ação**. [Derek, L. 2008]

## Objetivos de aprendizagem

- Saber identificar as ações de acoplar ou desacoplar as entidades ao contexto de persistência.
- Compreender os conceitos e métodos relacionados ao **EntityManager**.
- Diferenciar os tipos de contexto de persistência.

## Preparação para teste de leitura

1. O que é uma entidade desacoplada?  
R → Trata-se de uma entidade que não é gerenciada pelo EntityManager.
2. Quais os tipos de transações para gerenciarmos entidades?  
R →

## Teste de objetivos de aprendizagem

1. O que é contexto de persistência?  
R → É um gerenciamento dos dados que ficam associados a uma entidade.

2. Diferencie os tipos dos contexto de persistência?

R → O contexto de persistência pode ser gerenciado (Acoplado), quando o dados são geridos por um gerenciador de entidades, ou não gerenciado (Desacoplado) deste modo os dados não são mais atualizados no banco.

3. Construa um **quadro** com os principais métodos do **EntityManager** e sua respectiva descrição.

R →

getTransaction

- Cria uma transaction para poder gerir as transações.

persist

- Persiste novos dados no banco de dados.

merge

- Atualiza os dados no banco de dados.

remove

- Remove os dados do banco de dados.

contains

- Checa se a instância do Entity está em um estado persistente.

find

- Procura um registro do banco de dados através da entity e id da tabela, caso não encontre retorna null.

getReference

- Procura um registro no banco de dados através da entity e id da tabela, caso não encontre retorna uma exceção **javax.persistence.EntityNotFoundException**

flush

- Os métodos persist(), merge() e remove() aguardam finalização da transação para sincronizar as entidades com o banco de dados, o método flush() força essa sincronização no momento em que é chamado.

refresh

- Verifica se houve alguma alteração no banco de dados para sincronizar com a entidade.

clear

- Remove as entidades que estão no estado gerenciável dentro da EntityManager.

close

- Fecha a conexão do EntityManager.

isOpen

- Verifica se o EntityManager está com a conexão aberta.

4. Qual a diferença do método **merge** e **refresh**?

R → merge serve para atualizar os dados de um entity no banco de dados, o refresh verifica se houve alguma atualização no banco de dados para sincronizar com a entidade.

5. Desenvolva uma classe que implementa os métodos CRUD para uma entidade. (O ideal é seguir o formato padrão para os projetos da disciplinas).

R → <https://github.com/ifpb-disciplinas-2016-2/peerPersistencia.git>