

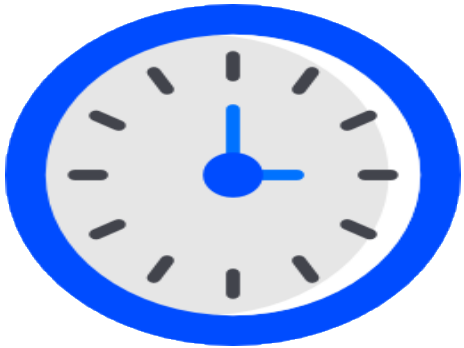
**FIA/P GRADUAÇÃO**



# JAVA ADVANCED

## #03 - AGENDA

---



- Entity Manager
- Persistence Unit e Persistence Context
- Obter um Entity Manager
- Controlar as transações
- Estados de uma Entidade
- Métodos do Entity Manager:
  - Persist
  - Find
  - Merge
  - Refresh
  - Remove
- Métodos do ciclo de vida da Entidade



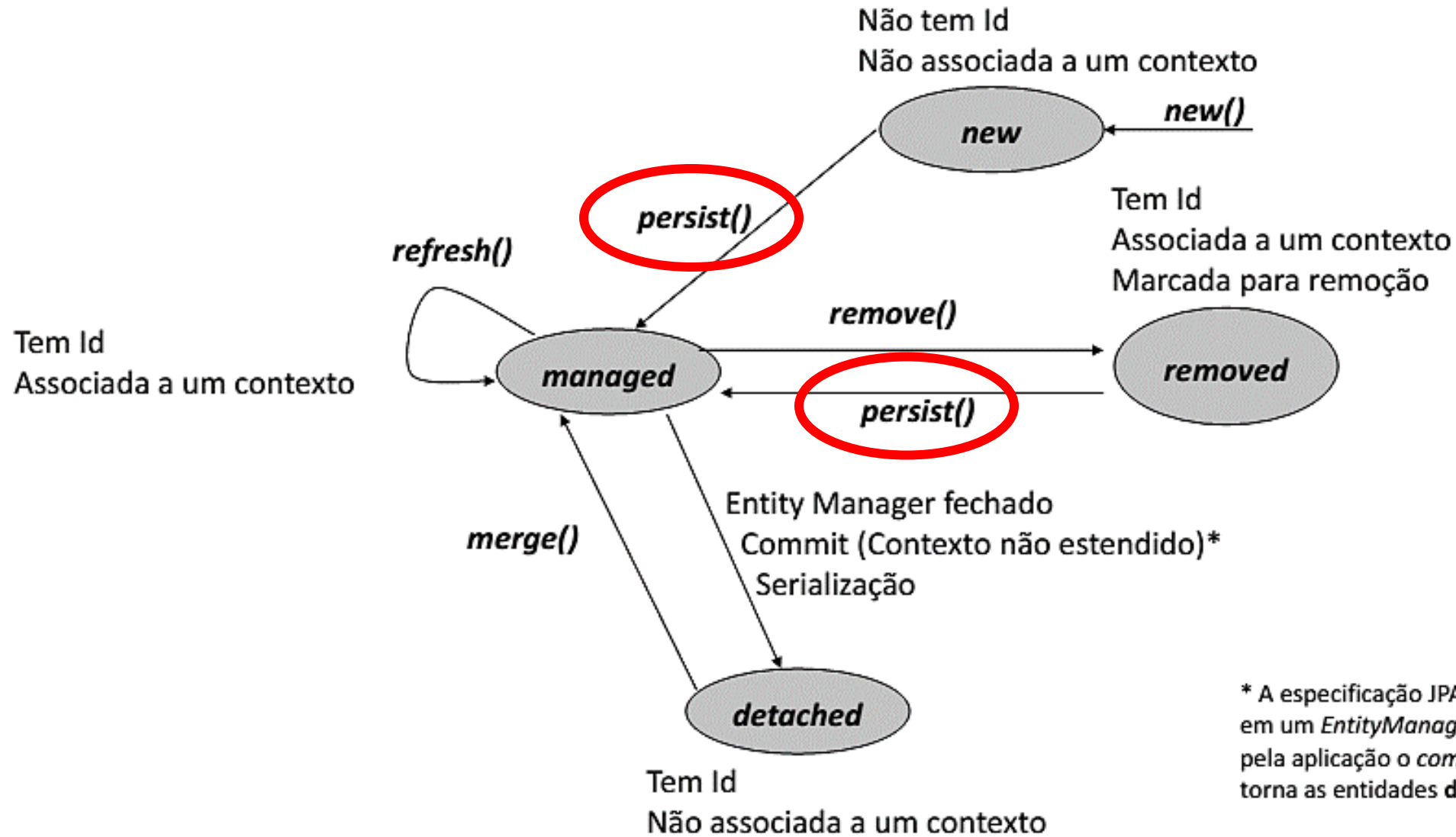
# MÉTODOS DO ENTITY MANAGER

- **persist(Object entity)**: enfileira uma nova entidade para ser inserida no banco de dados e a torna gerenciada;
  - ✓ Caso a entidade seja **NEW** então torna-se **MANAGED**;
  - ✓ Caso a entidade seja **MANAGED** ela é ignorada;
  - ✓ Caso a entidade seja **REMOVED** então torna-se **MANAGED**;
  - ✓ Caso a entidade seja **DETACHED**, uma **IllegalArgumentException** é lançada;

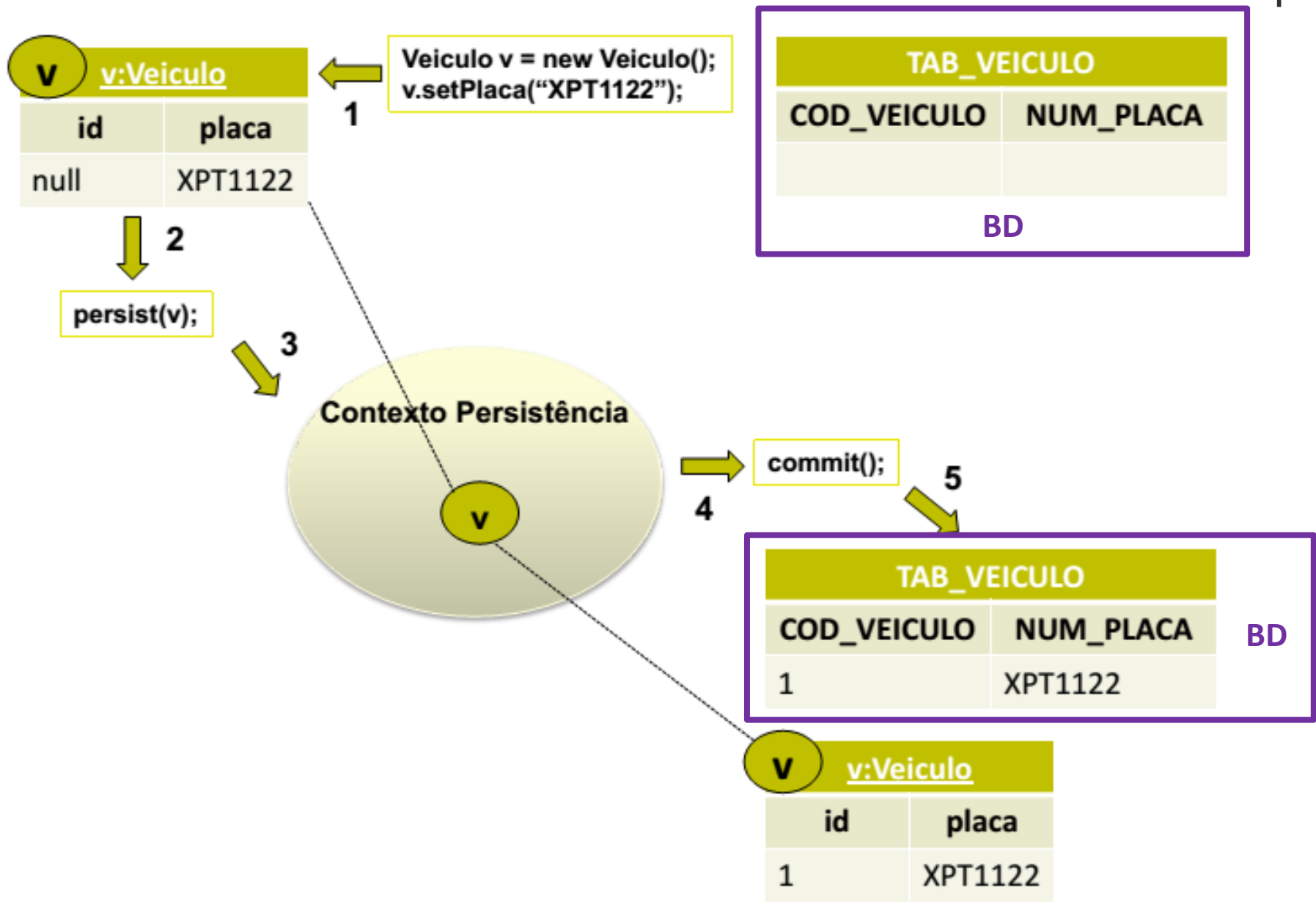
```
Veiculo veiculo = new Veiculo();
veiculo.setPlaca("DHZ-5678");
veiculo.setModelo("Gol");

manager.persist(veiculo);
```





\* A especificação JPA define que em um *EntityManager* gerenciado pela aplicação o *commit* não torna as entidades **detached**.

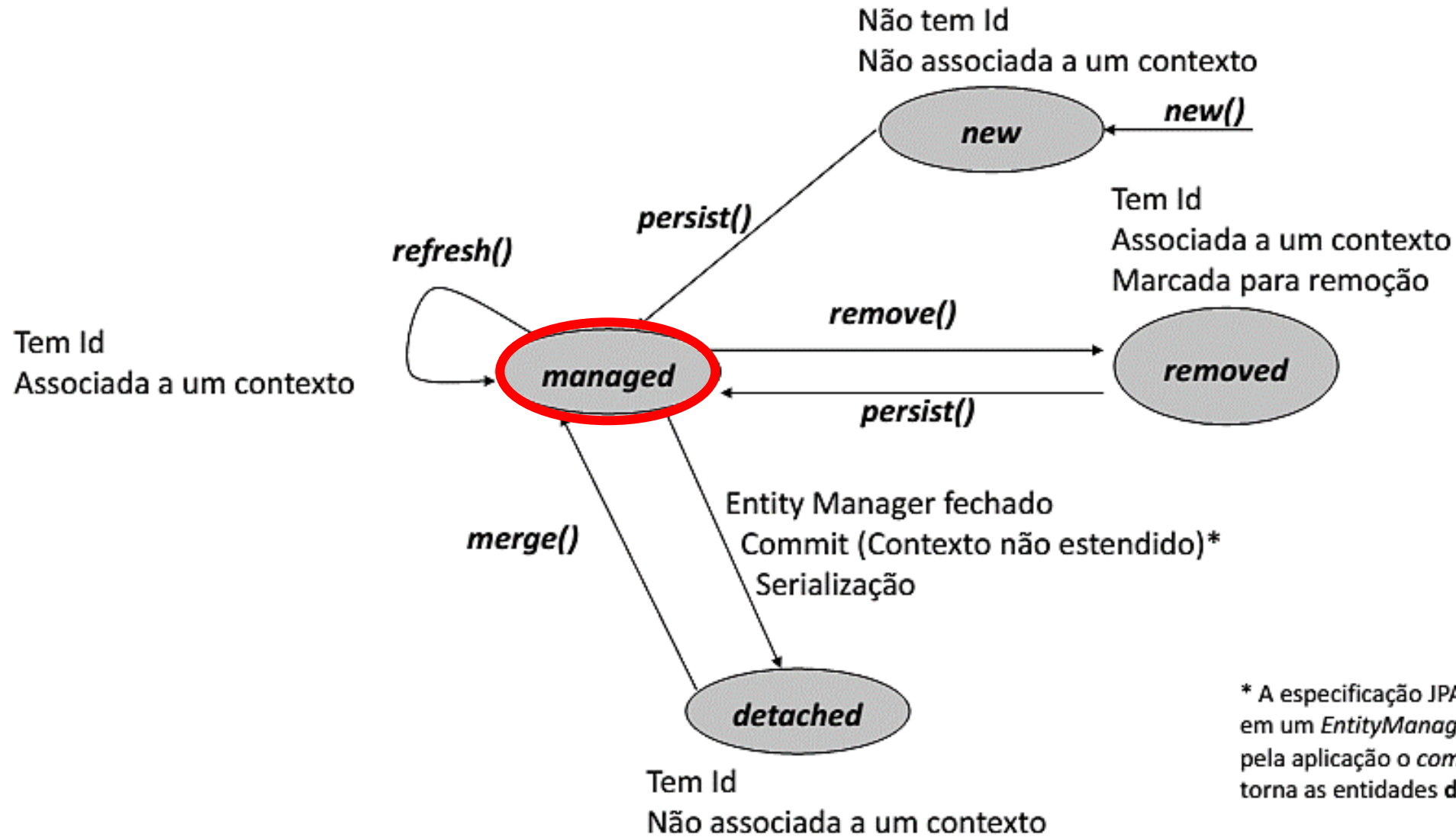


- `<T> T find (Class<T> classeEntidade, Object PK)`: realiza uma busca por meio da chave primária da entidade;
  - ✓ Retorna uma instância **MANAGED** caso seja localizada ou **null** caso contrário;

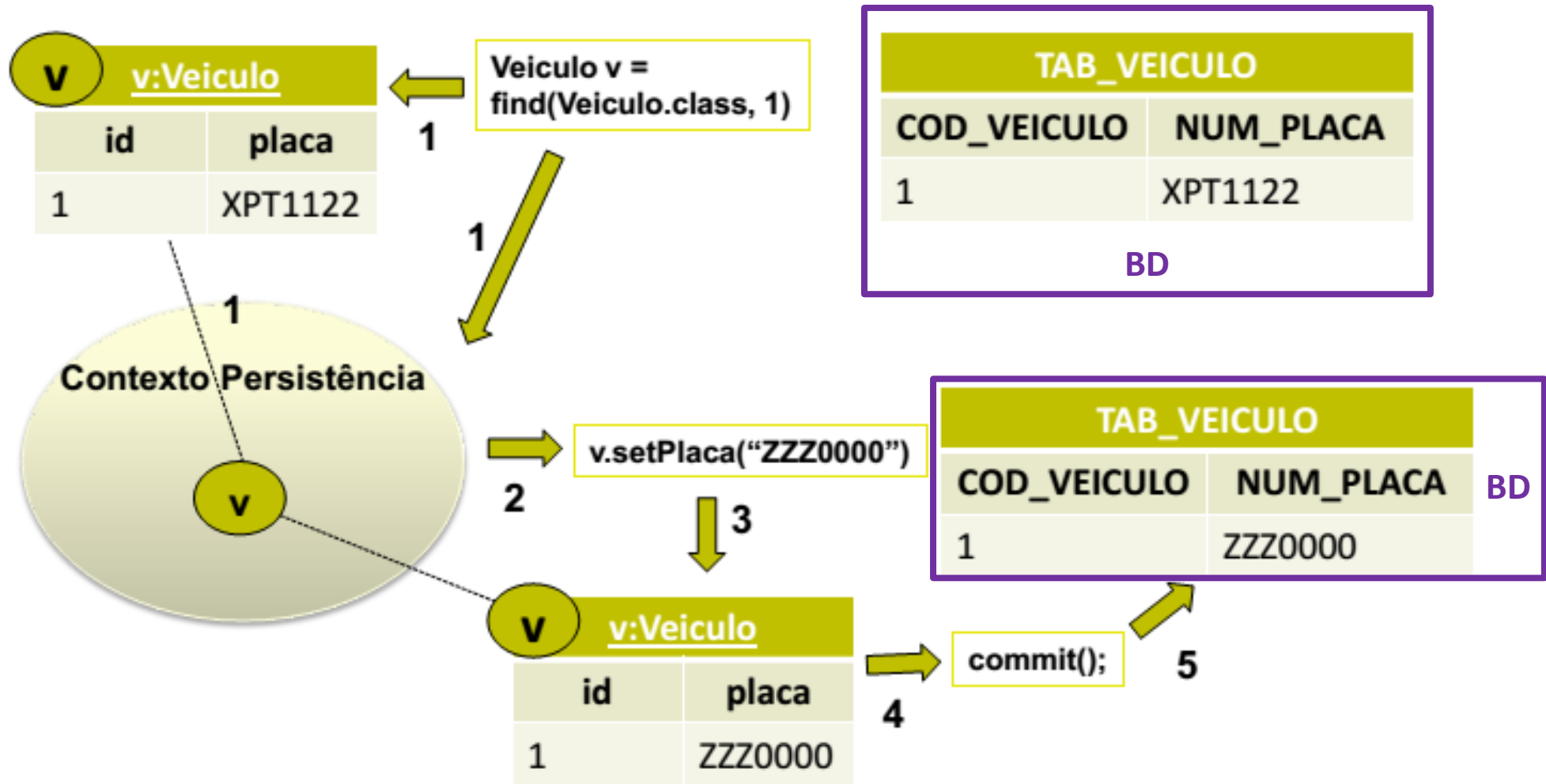
```
// Busca veiculo com id igual a 10  
Veiculo veiculo = manager.find(Veiculo.class, 10);
```







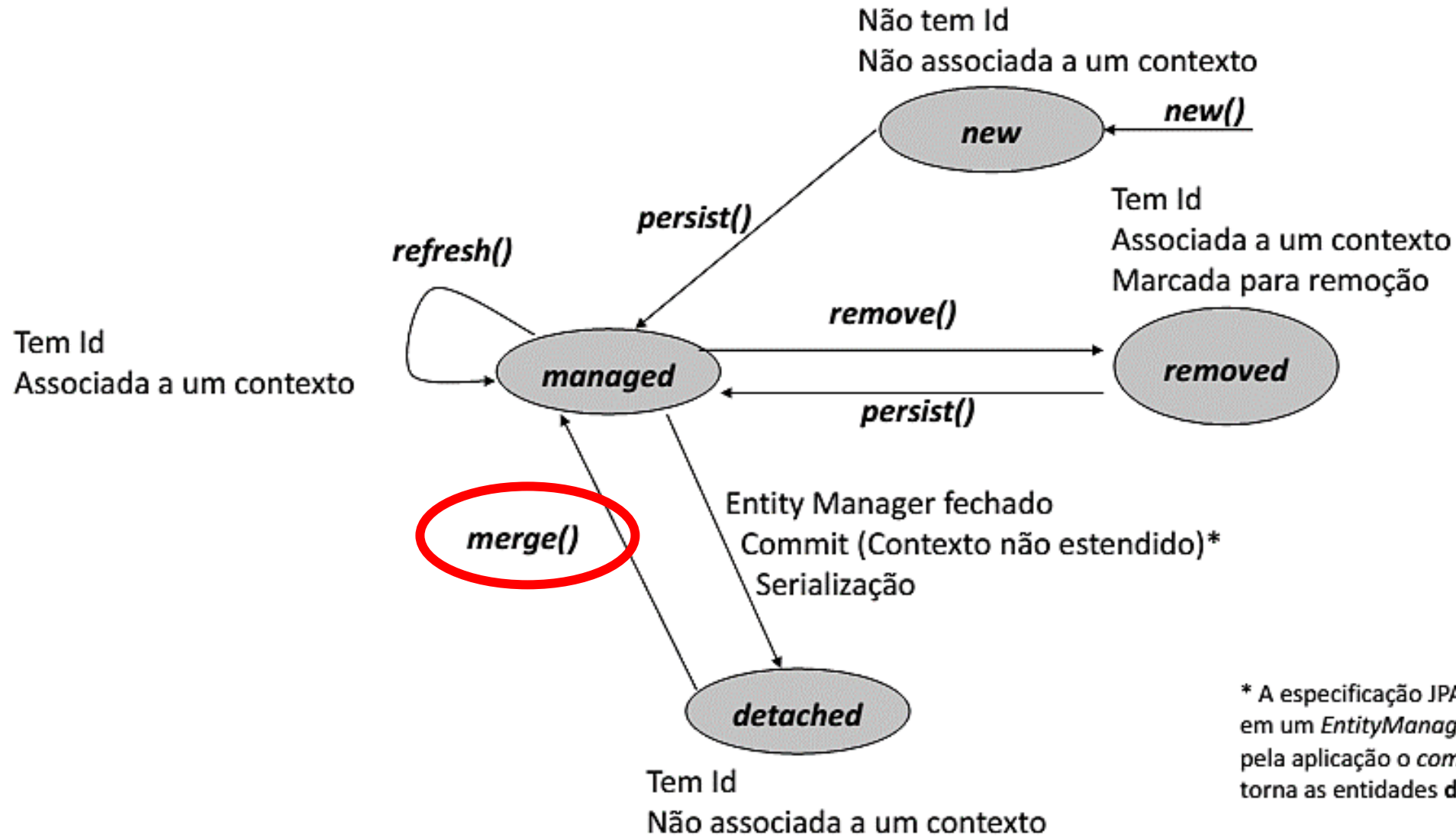
\* A especificação JPA define que em um *EntityManager* gerenciado pela aplicação o *commit* não torna as entidades **detached**.



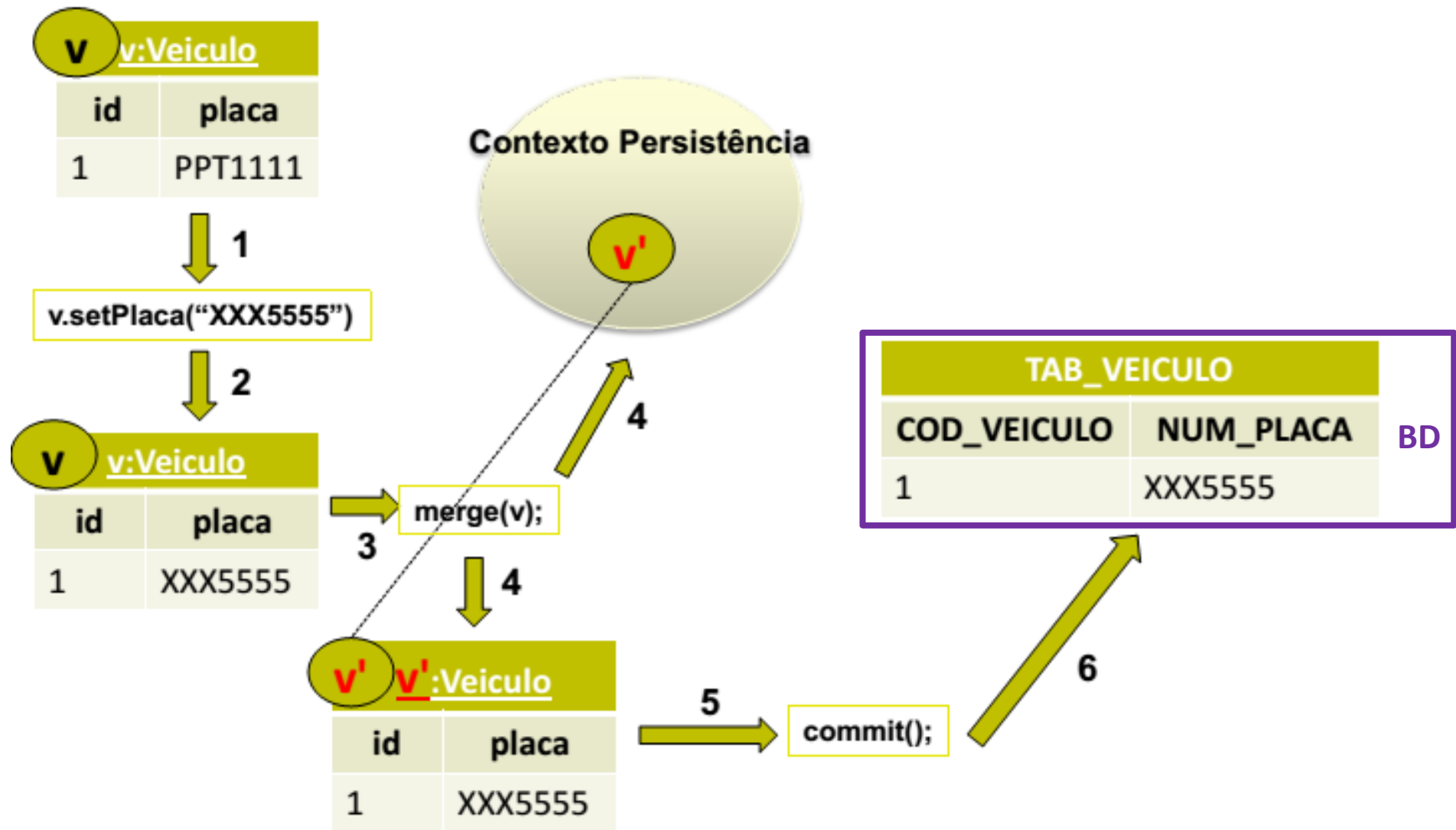
- **<T> T merge (T entidade):** retorna uma cópia gerenciada de uma entidade não gerenciada;
  - ✓ Caso a entidade seja **DETACHED**, seu estado é copiado para uma instância **MANAGED** com a mesma identidade (ID) ou uma nova cópia **MANAGED** da entidade é criada;
  - ✓ Caso a entidade seja **NEW**, uma nova entidade **MANAGED** é criada com o estado copiado da entidade original;
  - ✓ Caso a entidade seja **MANAGED** ela é ignorada;
  - ✓ Caso a entidade seja **REMOVED**, uma **IllegalArgumentException** é lançada;

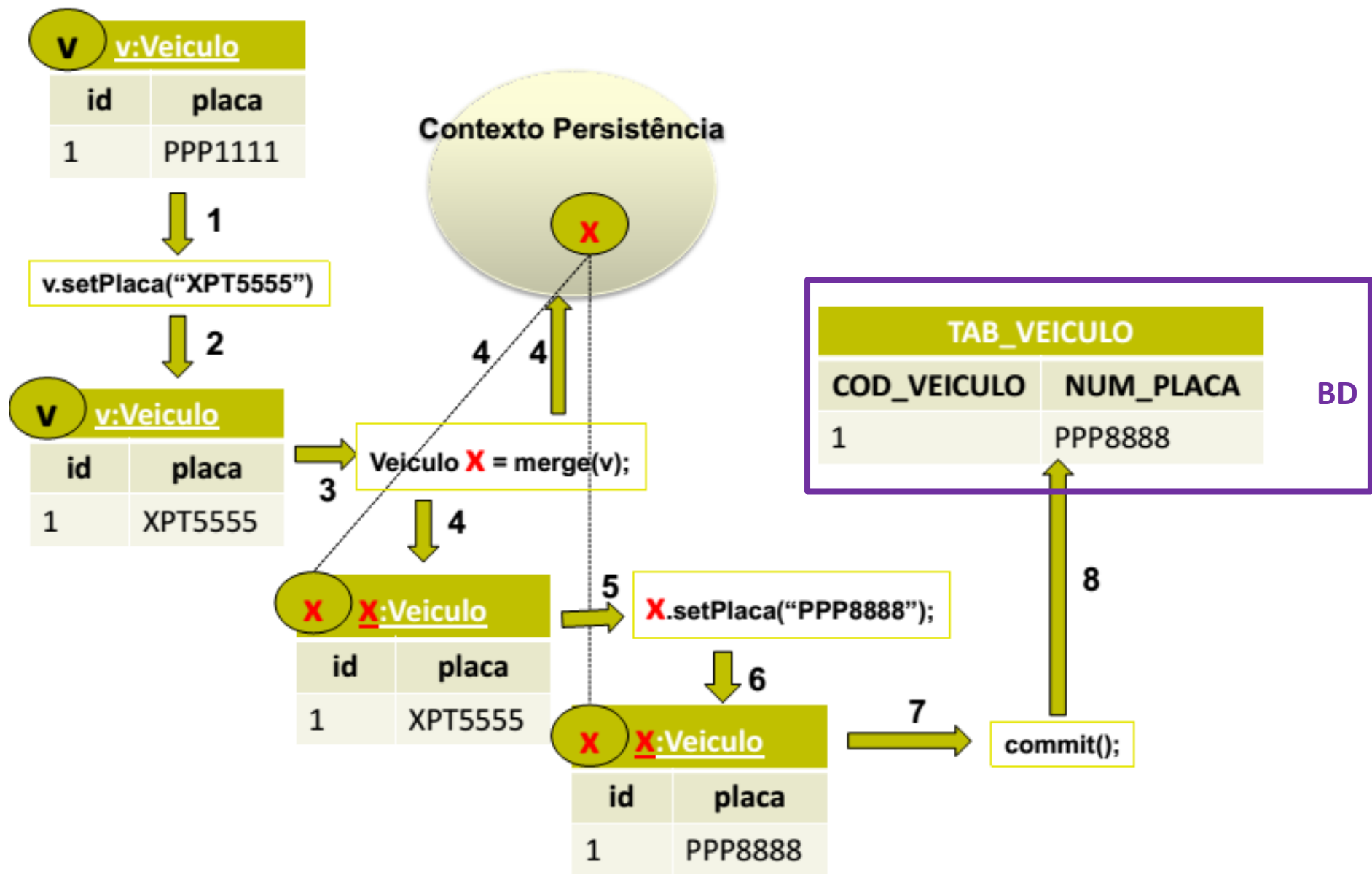
```
Veiculo v2 = manager.merge(veiculo);  
v2.setPlaca("DHZ-5678");  
v2.setModelo("Fusca");
```





\* A especificação JPA define que em um *EntityManager* gerenciado pela aplicação o *commit* não torna as entidades **detached**.

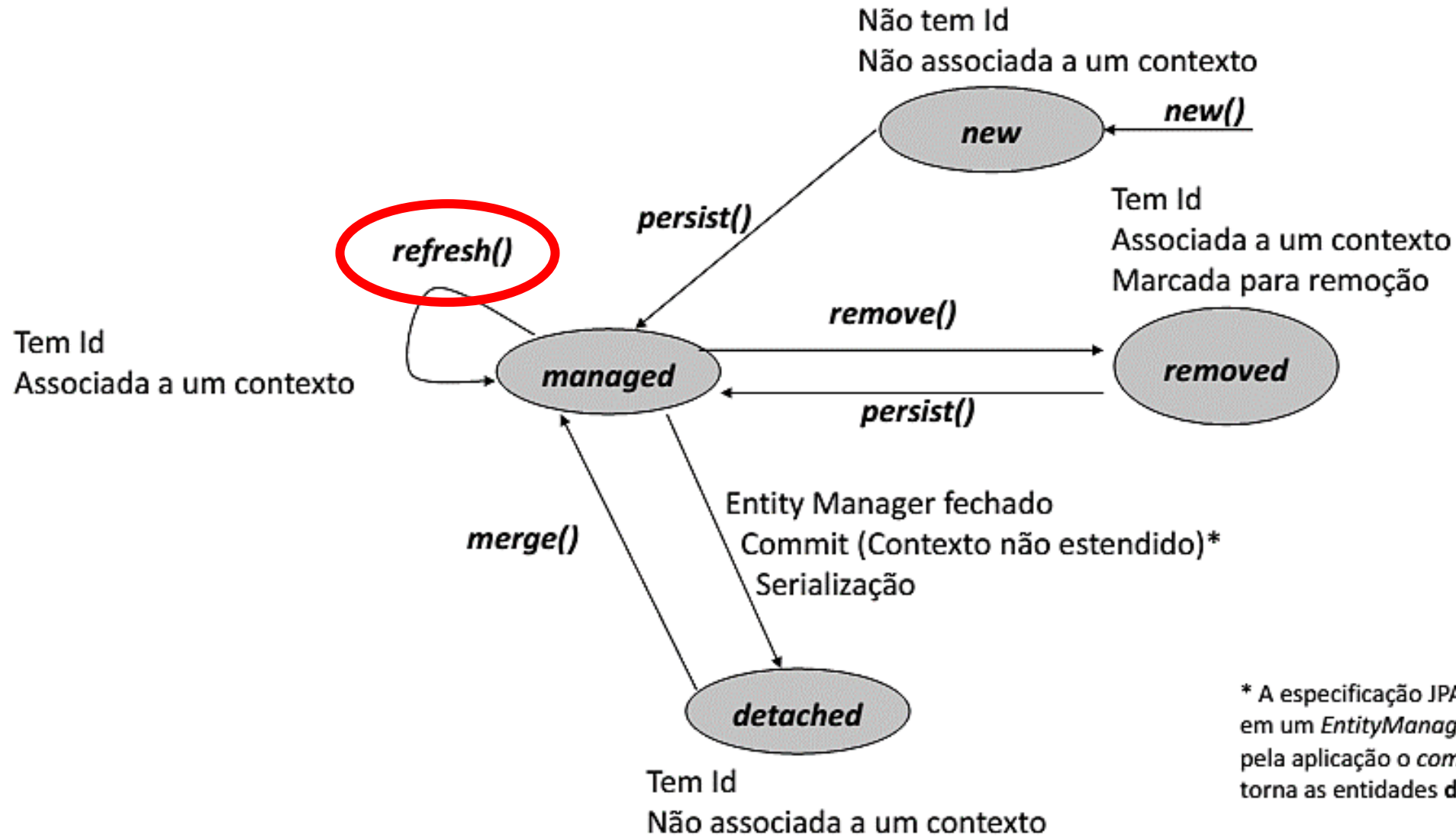




- **refresh(Object entidade)**: certifica que o estado da entidade encontra-se sincronizado com a base de dados;
  - ✓ Caso a entidade seja **NEW** ela é ignorada;
  - ✓ Caso a entidade seja **MANAGED**, seu estado é atualizado com a base de dados;
  - ✓ Caso a entidade seja **REMOVED**, ela é ignorada;
  - ✓ Caso a entidade seja **DETACHED**, uma **IllegalArgumentException** é lançada;

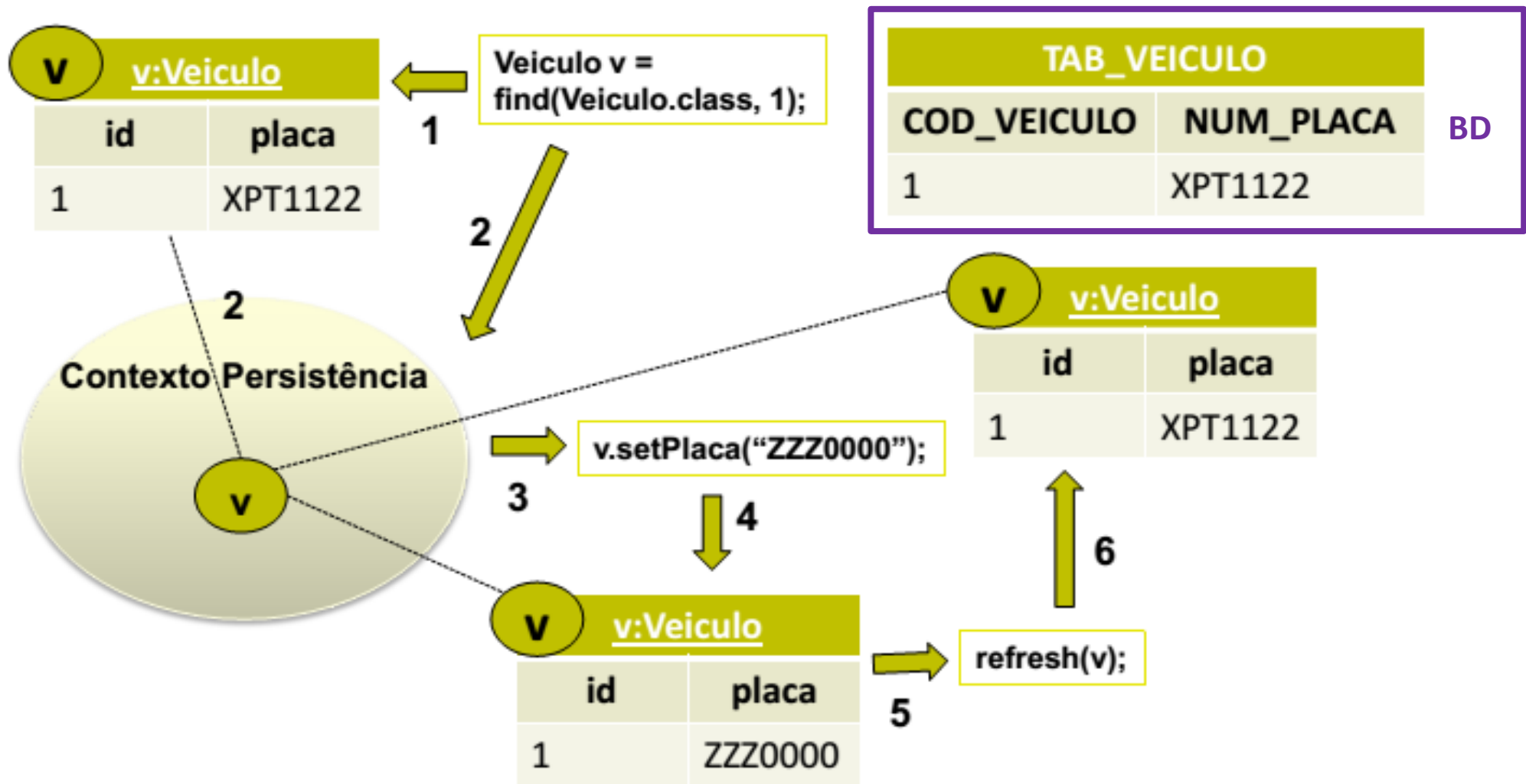
```
Veiculo veiculo = new Veiculo();  
veiculo = manager.find(Veiculo.class, 1);  
veiculo.setPlaca("DHZ-5678");  
veiculo.setModelo("Fusca");  
manager.refresh(veiculo);
```





\* A especificação JPA define que em um *EntityManager* gerenciado pela aplicação o *commit* não torna as entidades **detached**.

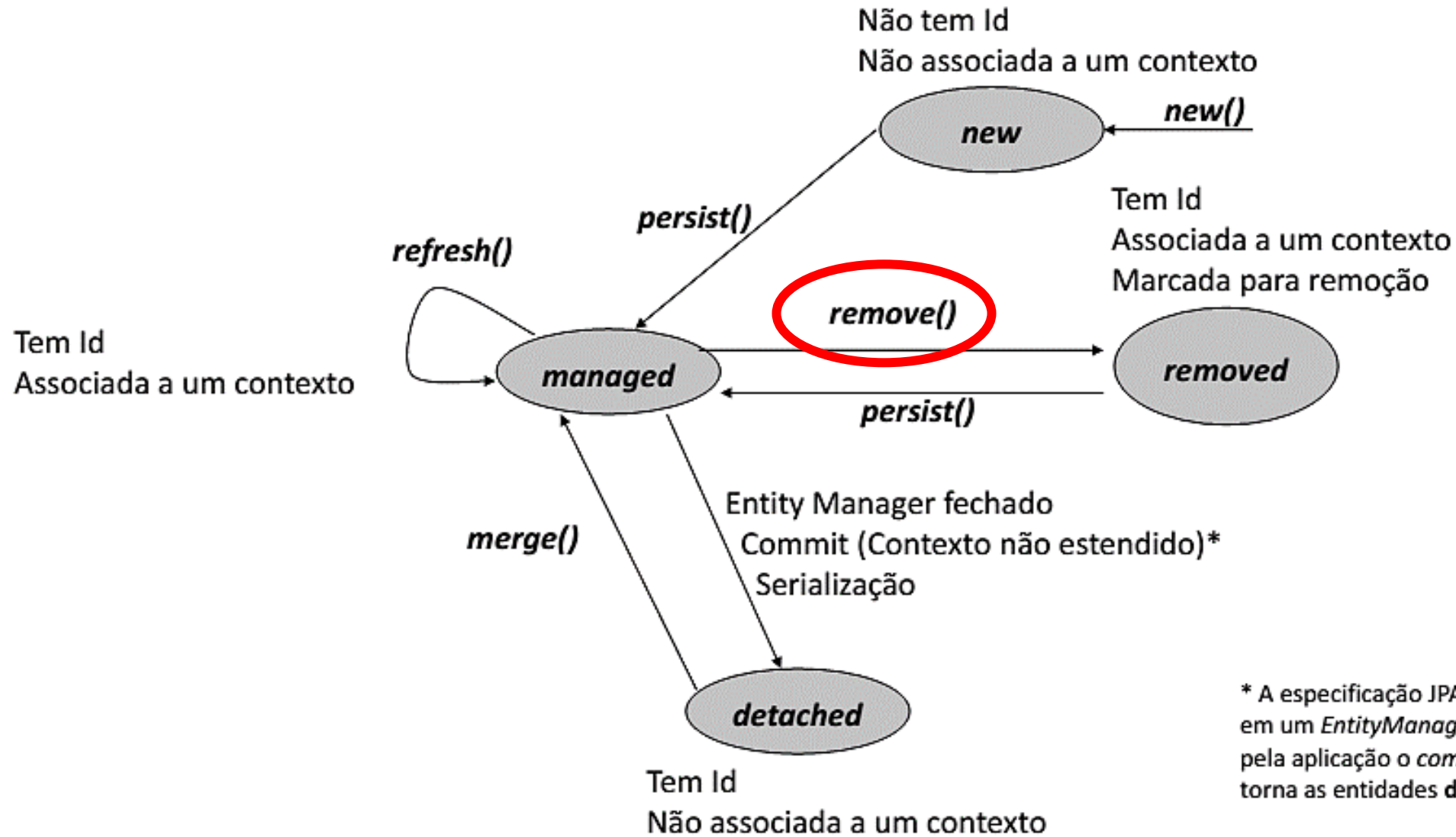




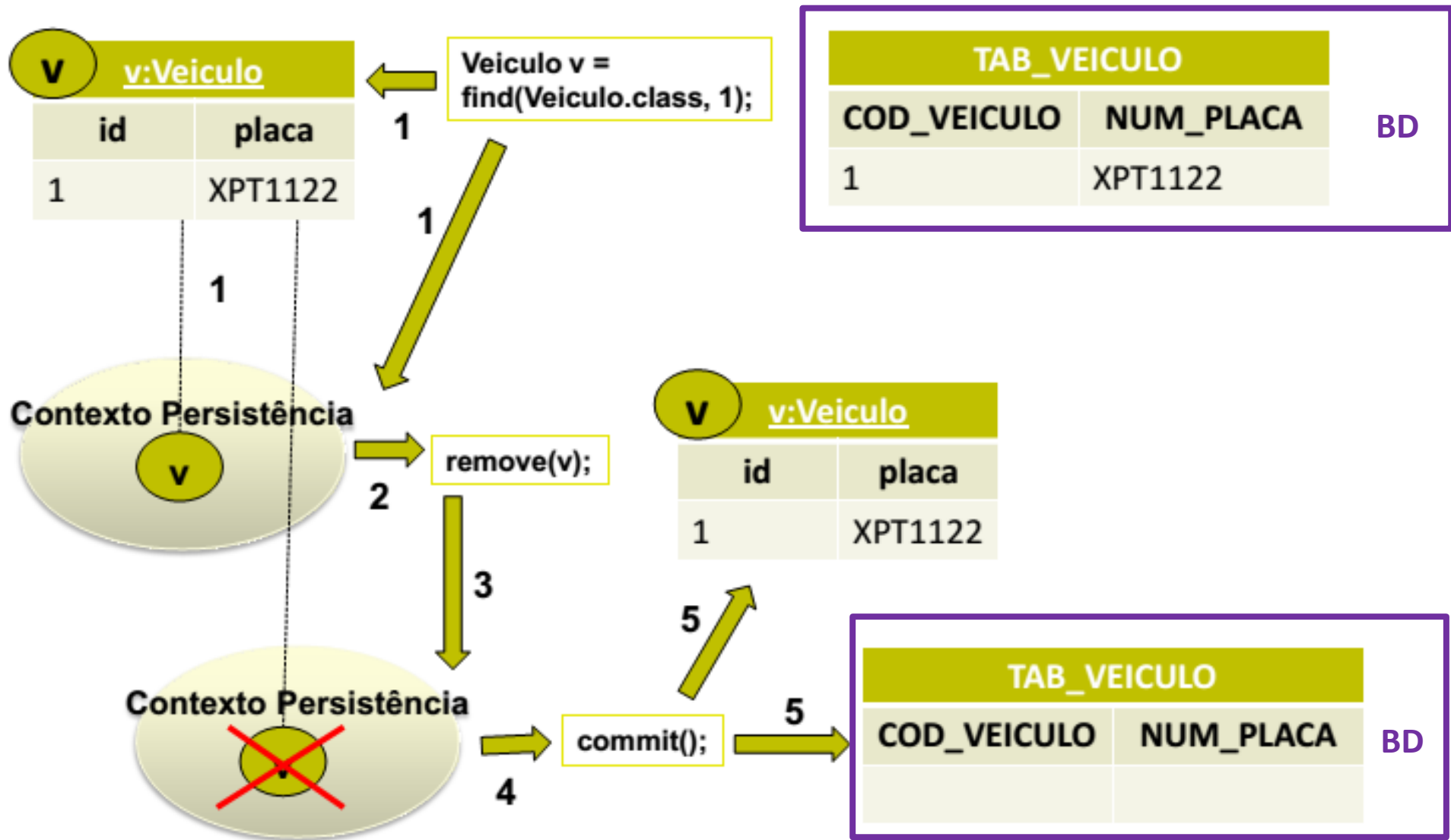
- **remove(Object entidade)**: marca uma entidade como **REMOVED** que será excluída do banco de dados após uma instrução **commit**;
  - ✓ Caso a entidade seja **NEW**, ela é ignorada;
  - ✓ Caso a entidade seja **MANAGED**, ela torna-se **REMOVED**;
  - ✓ Caso a entidade seja **REMOVED** ela é ignorada;
  - ✓ Caso a entidade seja **DETACHED**, uma **IllegalArgumentException** é lançada;

Para excluir entidades desacopladas primeiro deve-se torná-la **MANAGED**, por exemplo, utilizando o método **find**;

```
Veiculo veiculo = manager.find(Veiculo.class, 10);  
manager.remove(veiculo);
```



\* A especificação JPA define que em um *EntityManager* gerenciado pela aplicação o *commit* não torna as entidades **detached**.



- Pode-se realizar ações em cada fase do ciclo de vida de uma entidade;
- Para tanto, basta utilizar as anotações abaixo:
  - ✓ **@PrePersist**
  - ✓ **@PostPersist**
  - ✓ **@PreRemove**
  - ✓ **@PostRemove**
  - ✓ **@PreUpdate**
  - ✓ **@PostUpdate**
  - ✓ **@PostLoad**

# VOCÊ APRENDEU...

---



- O que é o **Entity Manager**, **Persistence Unit** e **Persistence Context**;
- Obter uma **instância** do Entity Manager e a controlar as **transações**;
- Sobre os **estados** da **entidade** e os **métodos**:
  - Persist;
  - Merge;
  - Find;
  - Refresh;
  - Remove;
- Métodos do **ciclo de vida** da entidade;

**Copyright © 2024 – 2034**  
**Prof. Dr. Marcel Stefan Wagner**

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).

Agradecimentos: Prof. Me Gustavo Torres Custódio | Prof. Me. Thiago T. I. Yamamoto

*“Se a vida não ficar mais fácil, trate de ficar mais forte.”*