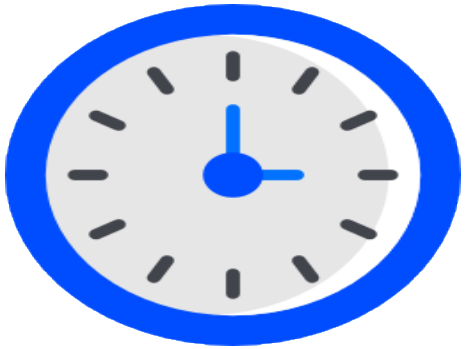
FIAP GRADUAÇÃO

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

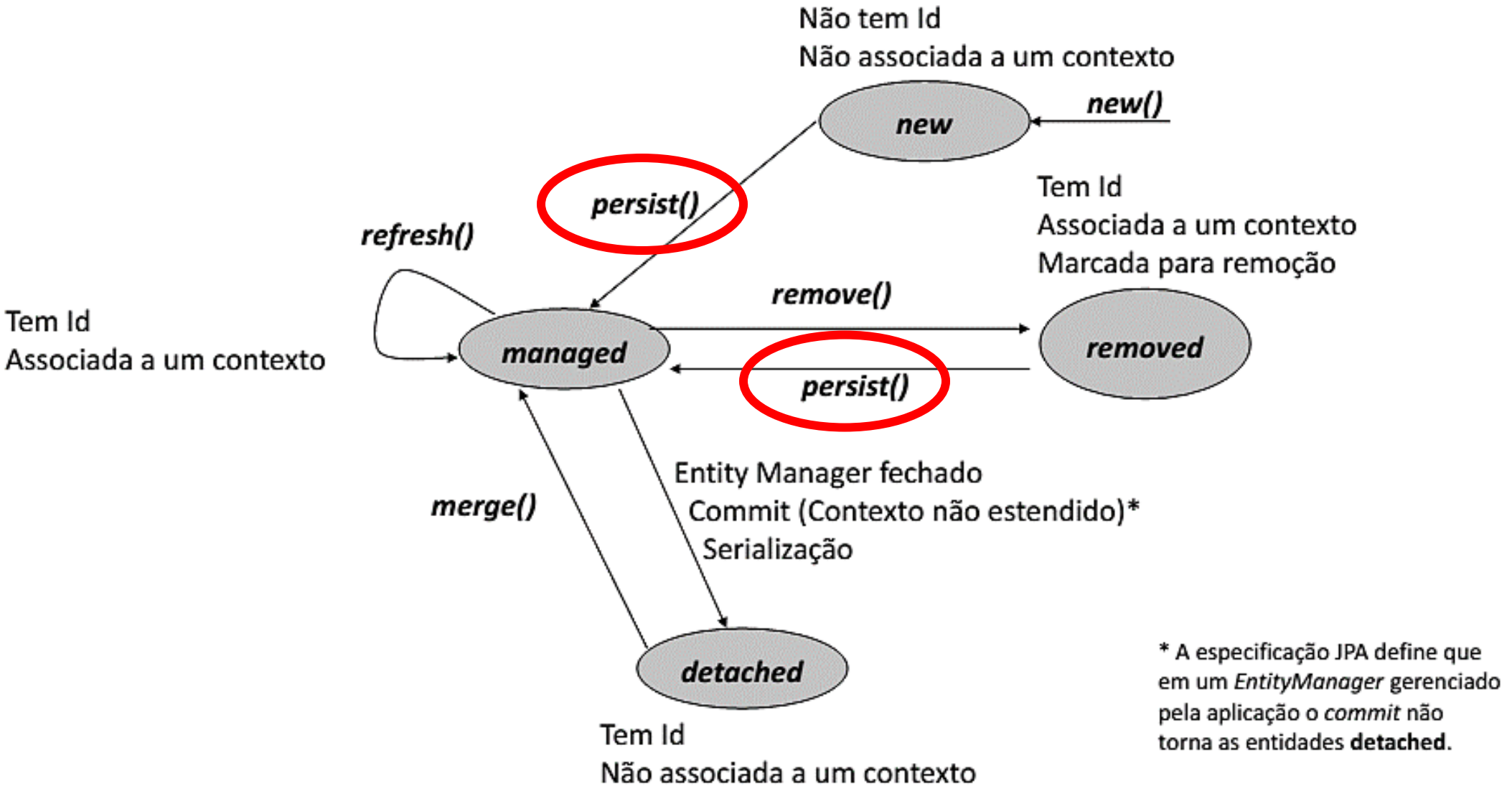
MÉTODOS DO ENTITY MANAGER

PERSIST

- **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;

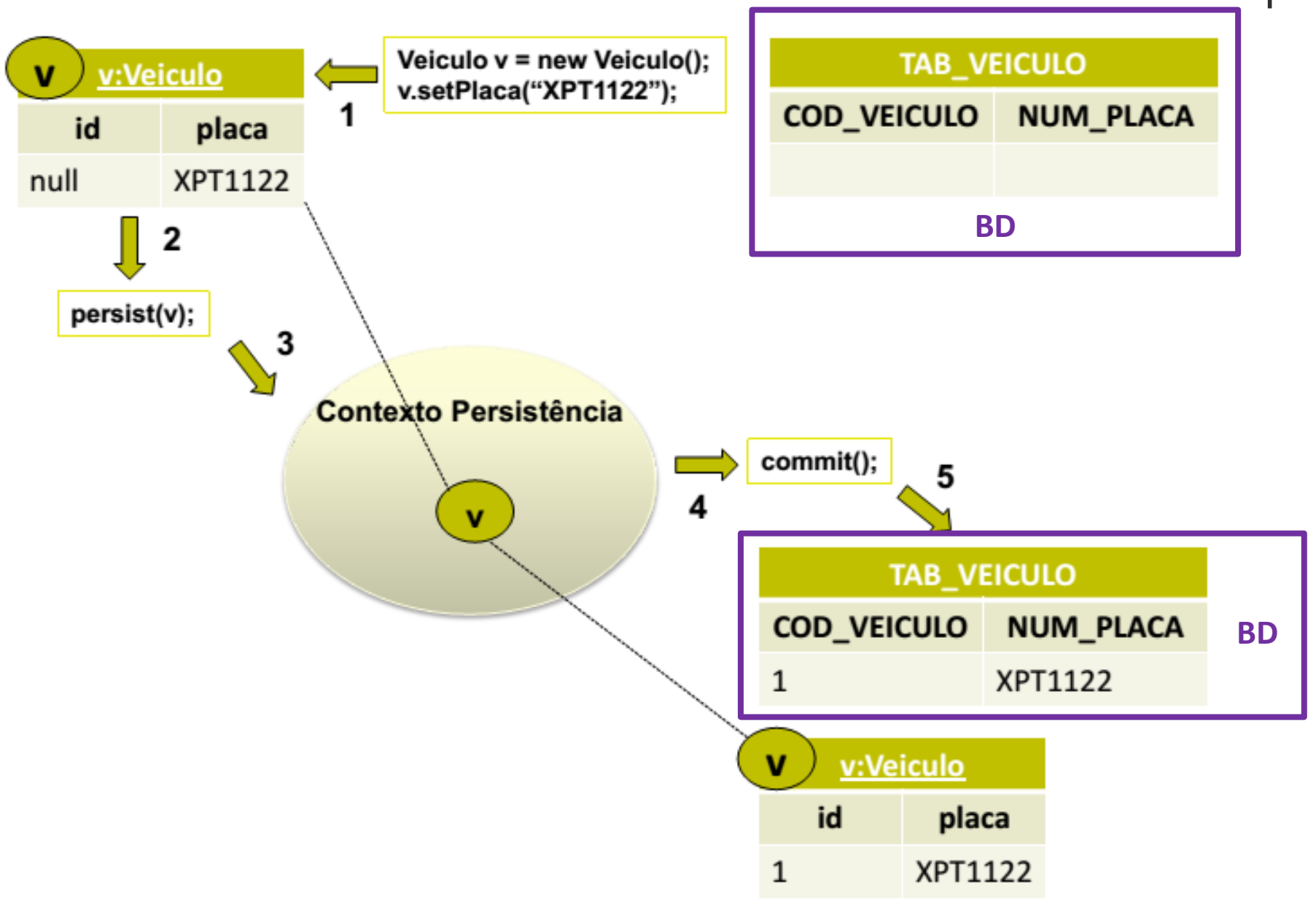
```
Veiculo veiculo = new Veiculo();
veiculo.setPlaca("DHZ-5678");
veiculo.setModelo(”Gol");

manager.persist(veiculo);
```

ESTADOS DA ENTIDADE



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

PERSIST

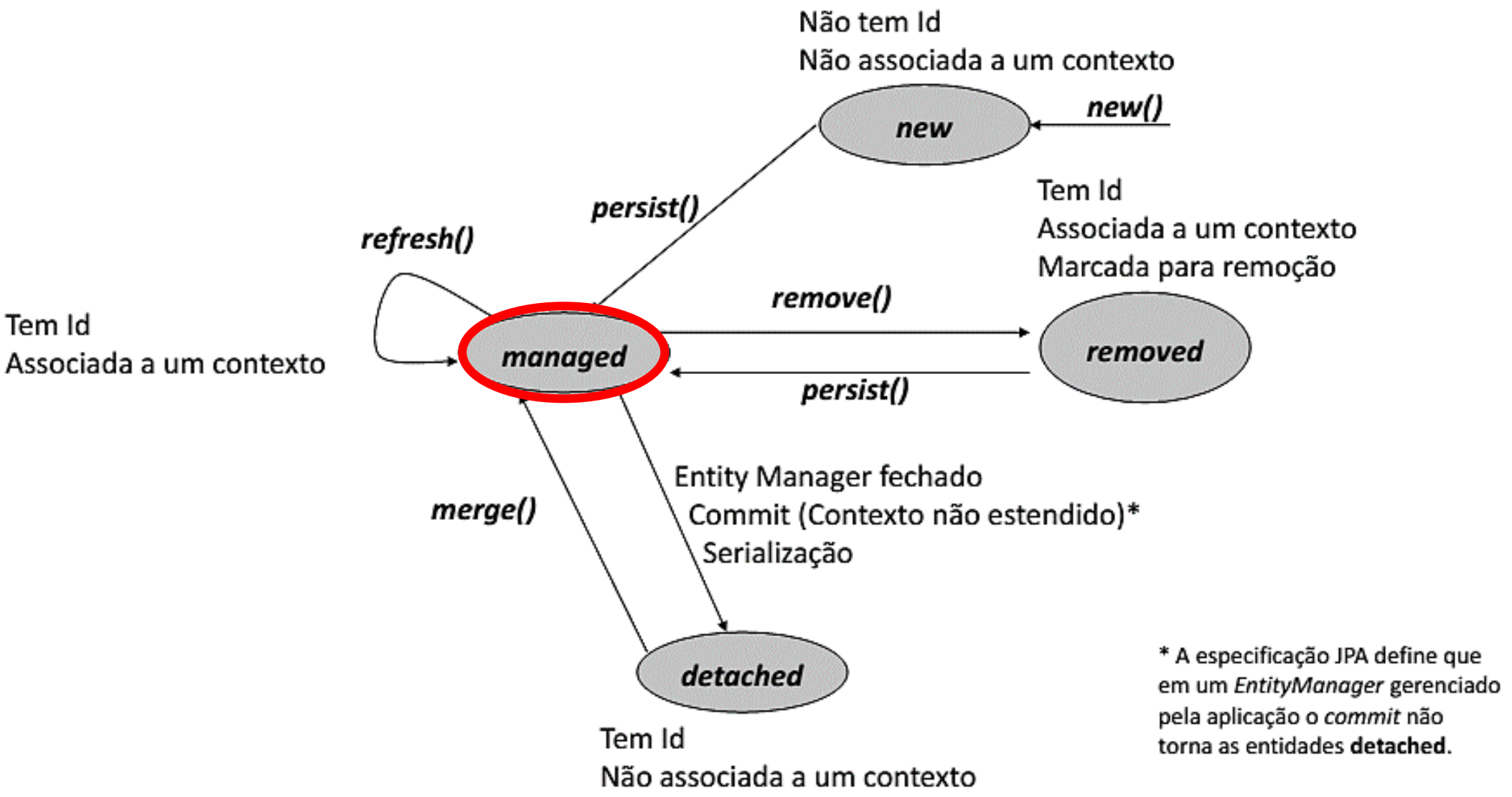

v
v:Veiculo
id
placa
null
XPT1122
Veiculo v = new Veiculo();
v.setPlaca("XPT1122");
1
2
persist(v);
3
Contexto Persistência
v
4
commit();
5
TAB_VEICULO
COD_VEICULO
NUM_PLACA
BD
TAB_VEICULO
COD_VEICULO
NUM_PLACA
1
XPT1122
BD
v
v:Veiculo
id
placa
1
XPT1122

FIND

- **<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;

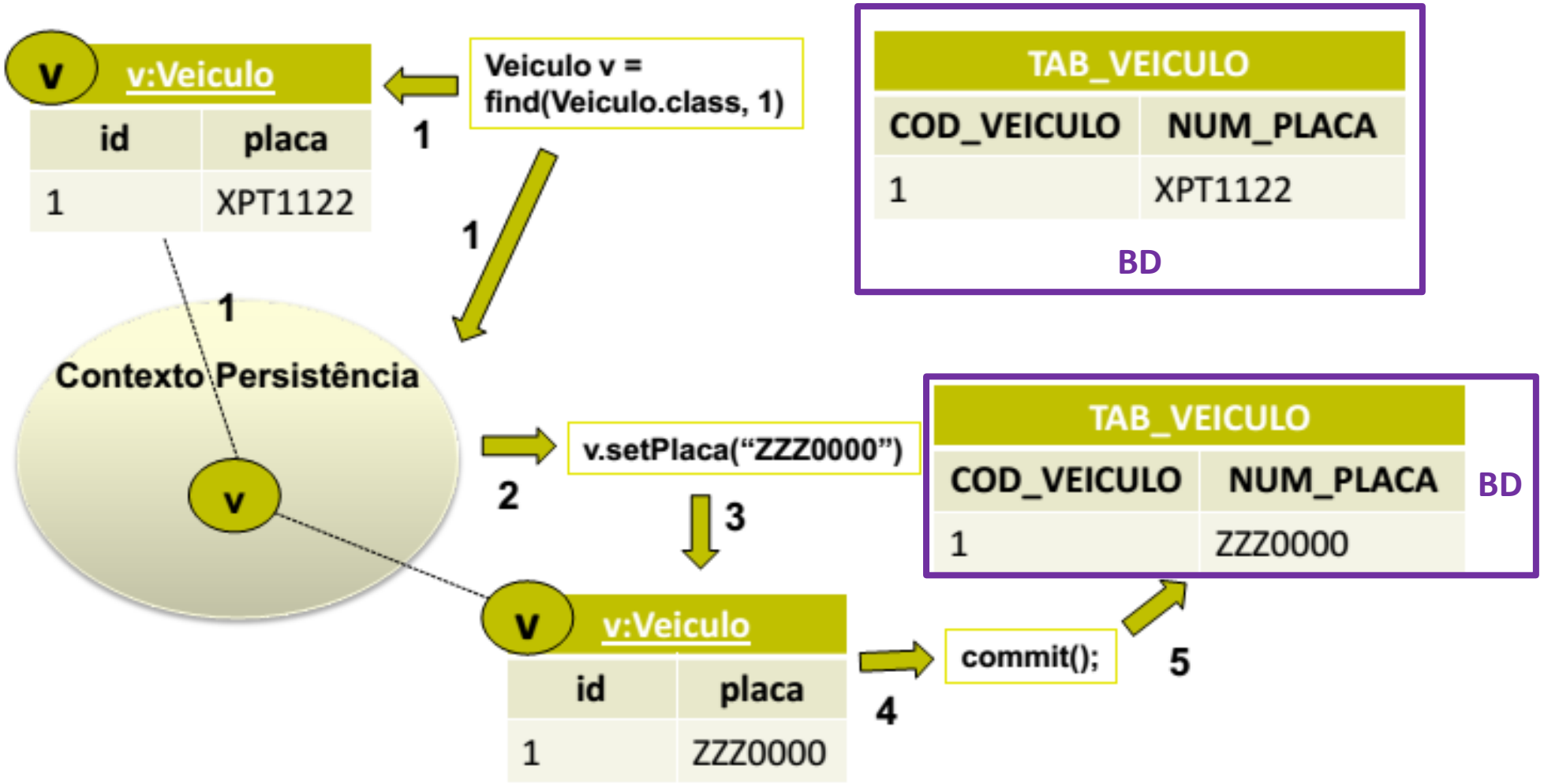
```
// Busca veiculo com id igual a 10
Veiculo veiculo = manager.find(Veiculo.class, 10);
```

ESTADOS DA ENTIDADE



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

FIND

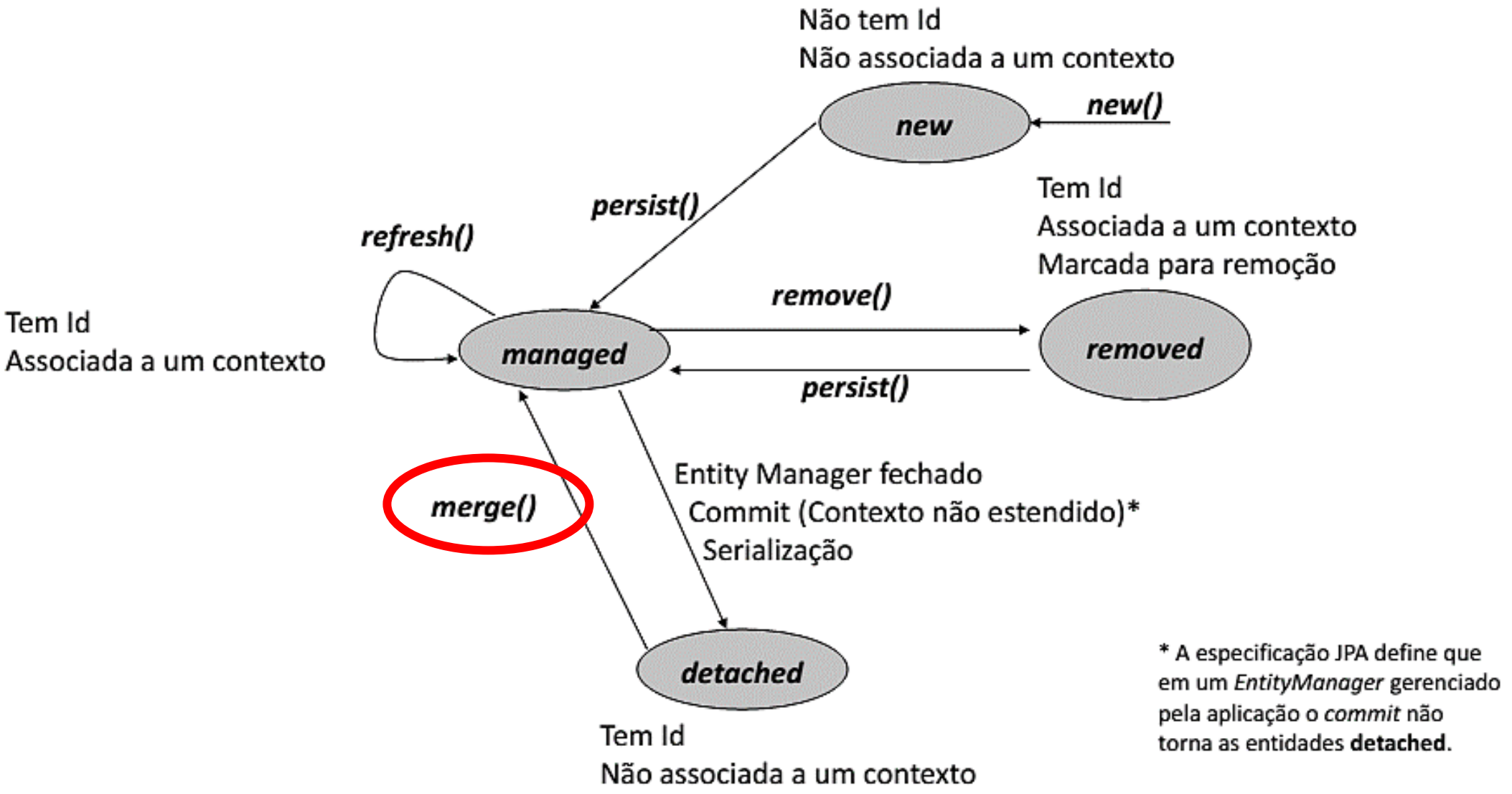

v
v:Veiculo
id
placa
1
XPT1122
Veiculo v = find(Veiculo.class, 1)
1
1
1
Contexto Persistência
v
2
v.setPlaca("ZZZ0000")
3
v
v:Veiculo
id
placa
1
ZZZ0000
4
commit();
5
TAB_VEICULO
COD_VEICULO
NUM_PLACA
1
XPT1122
BD
TAB_VEICULO
COD_VEICULO
NUM_PLACA
1
ZZZ0000
BD

MERGE

- **<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;

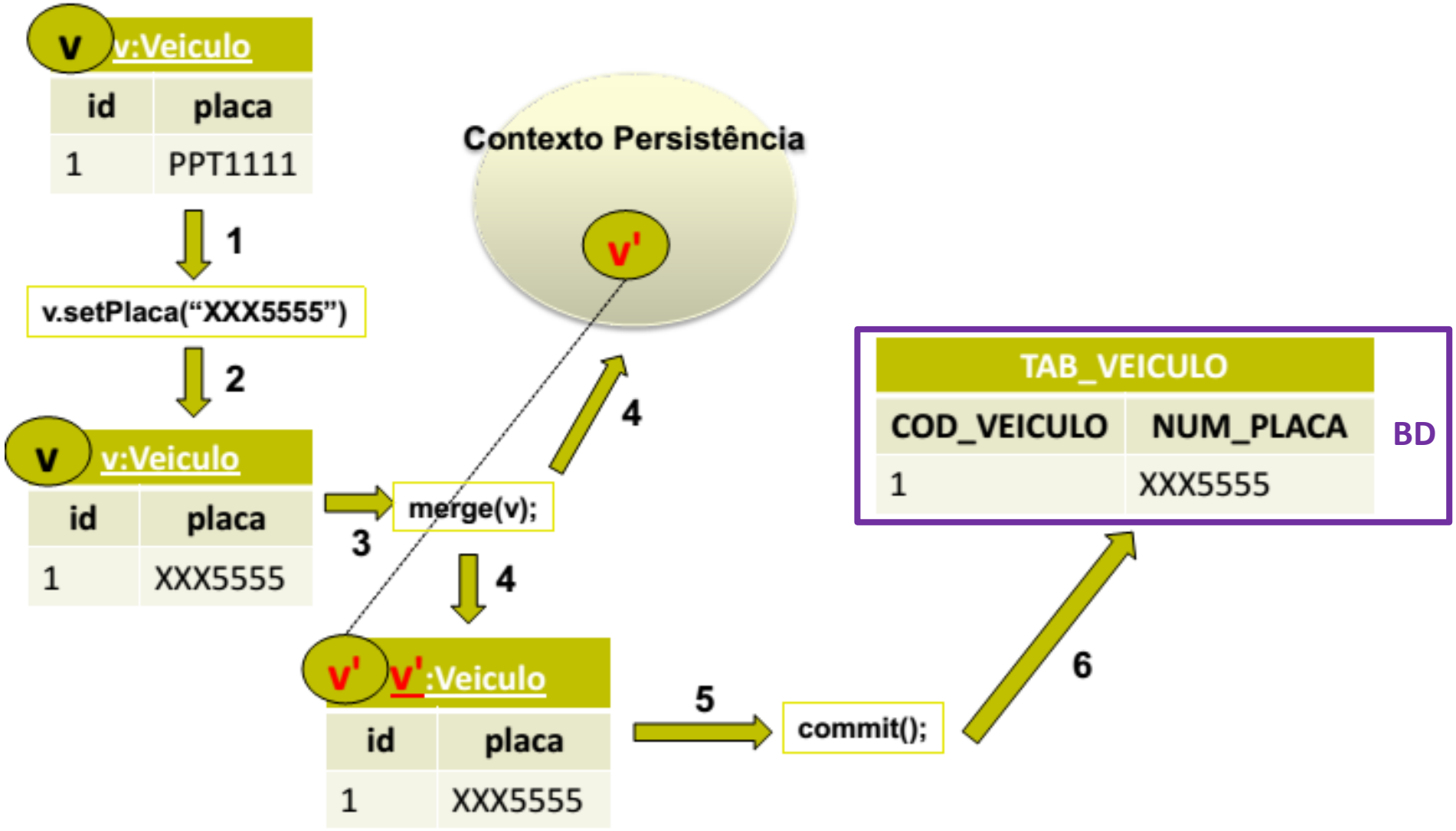
```
Veiculo v2 = manager.merge(veiculo);
v2.setPlaca("DHZ-5678");
v2.setModelo("Fusca");
```

ESTADOS DA ENTIDADE

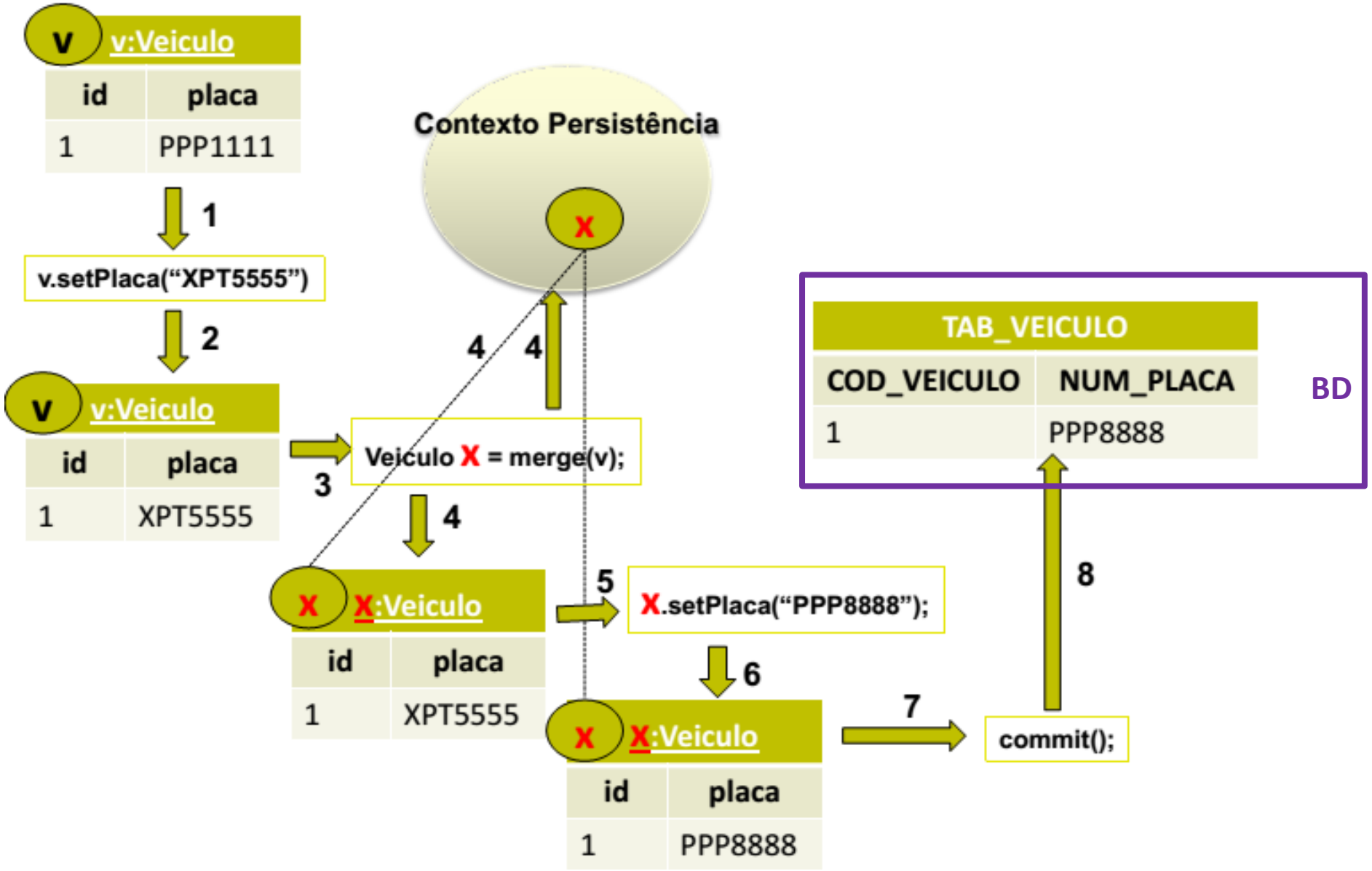


* A especificação JPA define que em um *EntityManager* gerenciado pela aplicação o *commit* não torna as entidades **detached**.

MERGE


v
v:Veiculo
id
placa
1
PPT1111
1
v.setPlaca("XXX5555")
2
v
v:Veiculo
id
placa
1
XXX5555
3
merge(v);
Contexto Persistência
v'
4
4
v'
v':Veiculo
id
placa
1
XXX5555
5
commit();
6
TAB_VEICULO
COD_VEICULO
NUM_PLACA
1
XXX5555
BD

MERGE

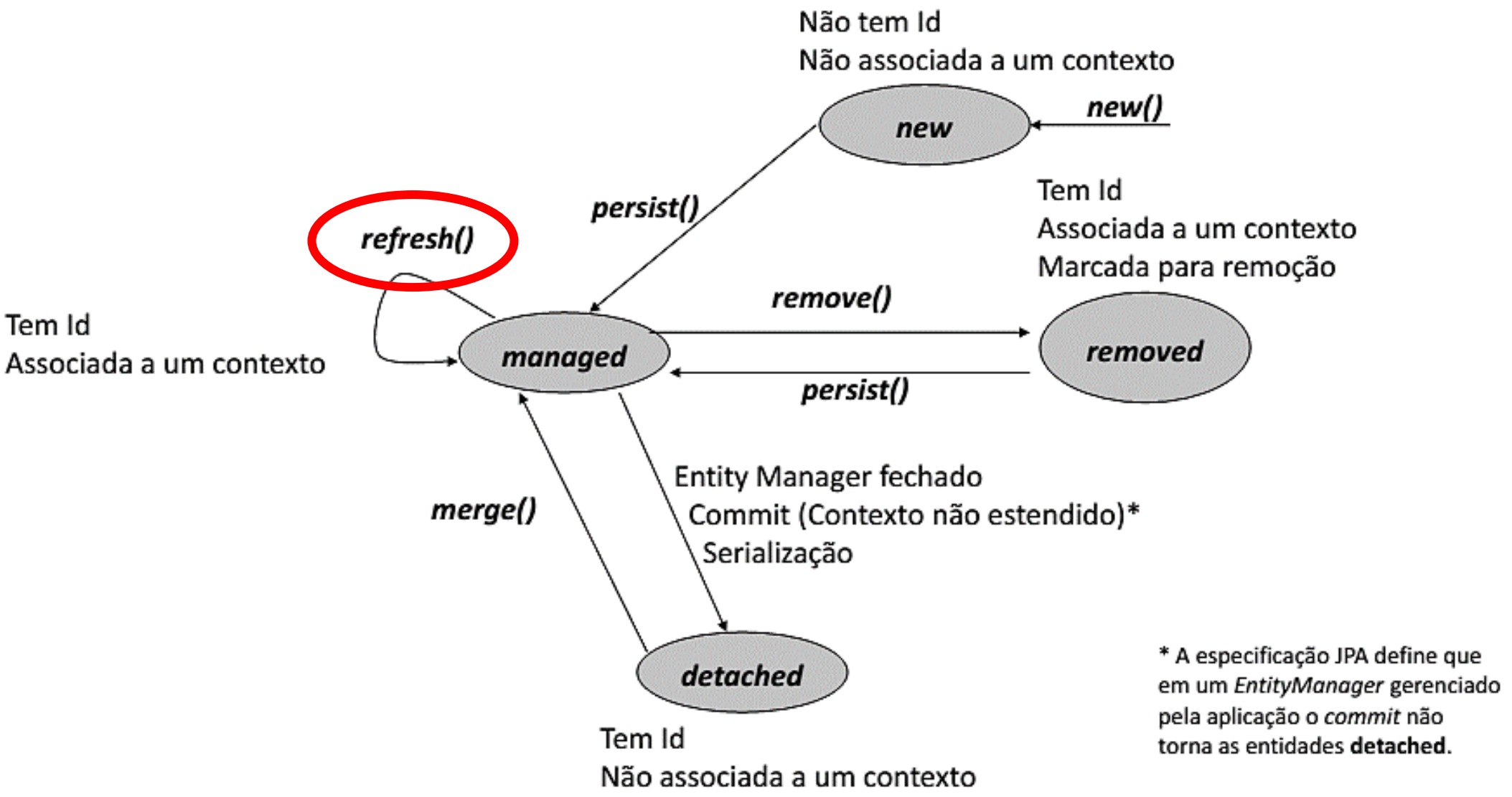

v
v:Veiculo
id
placa
1
PPP1111
1
v.setPlaca("XPT5555")
2
v
v:Veiculo
id
placa
1
XPT5555
Contexto Persistência
X
4
4
3
Veiculo X = merge(v);
4
X
X:Veiculo
id
placa
1
XPT5555
5
X.setPlaca("PPP8888");
6
X
X:Veiculo
id
placa
1
PPP8888
7
commit();
8
TAB_VEICULO
COD_VEICULO
NUM_PLACA
1
PPP8888
BD

REFRESH

- **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;

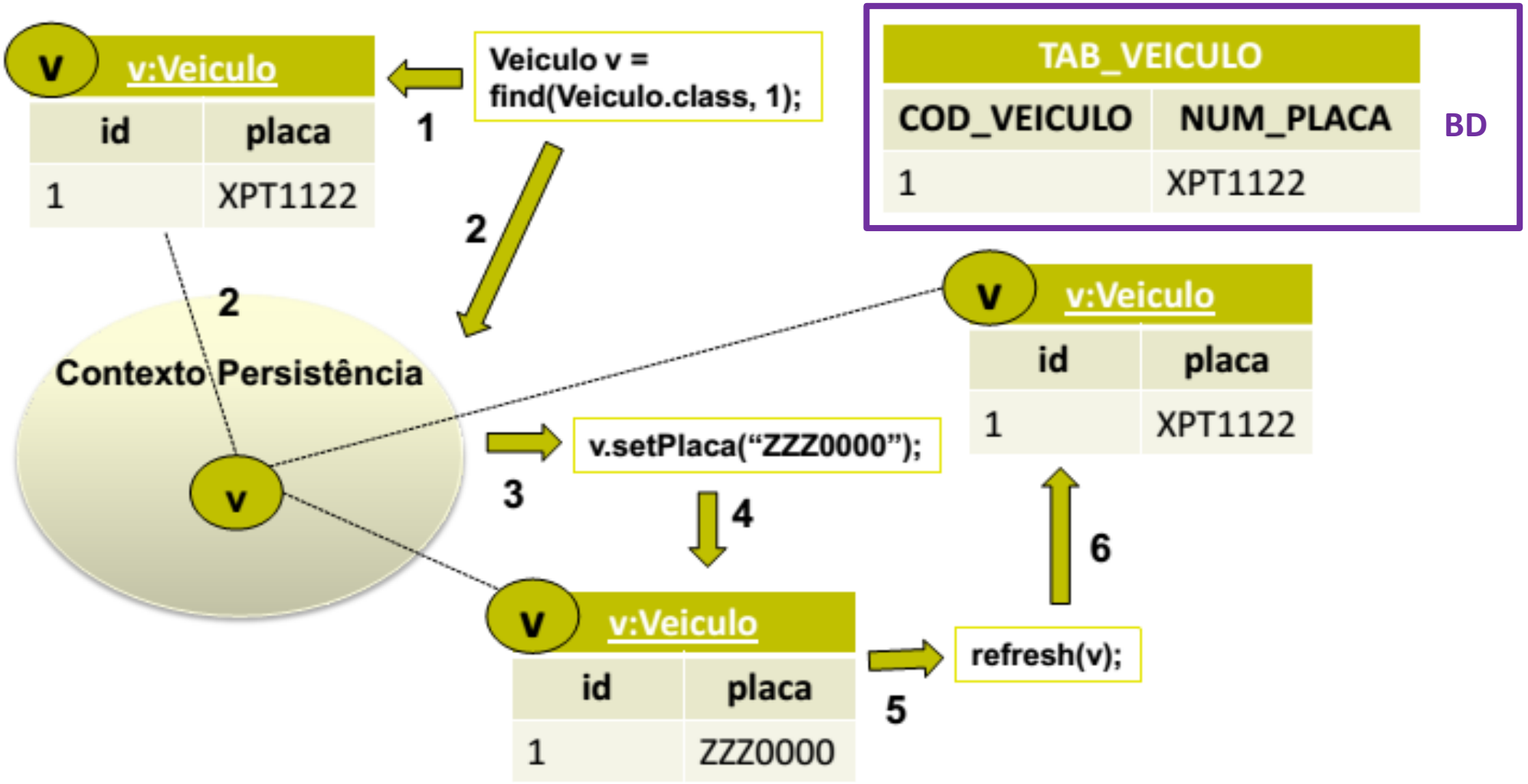
```
Veiculo veiculo = new Veiculo();
veiculo = manager.find(Veiculo.class, 1);
veiculo.setPlaca("DHZ-5678");
veiculo.setModelo("Fusca");
manager.refresh(veiculo);
```

ESTADOS DA ENTIDADE



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

REFRESH


v
v:Veiculo
id
placa
1
XPT1122
Veiculo v =
find(Veiculo.class, 1);
1
2
2
Contexto Persistência
v
TAB_VEICULO
COD_VEICULO
NUM_PLACA
1
XPT1122
BD
v
v:Veiculo
id
placa
1
XPT1122
v.setPlaca("ZZZ0000");
3
4
6
v
v:Veiculo
id
placa
1
ZZZ0000
5
refresh(v);

REMOVE

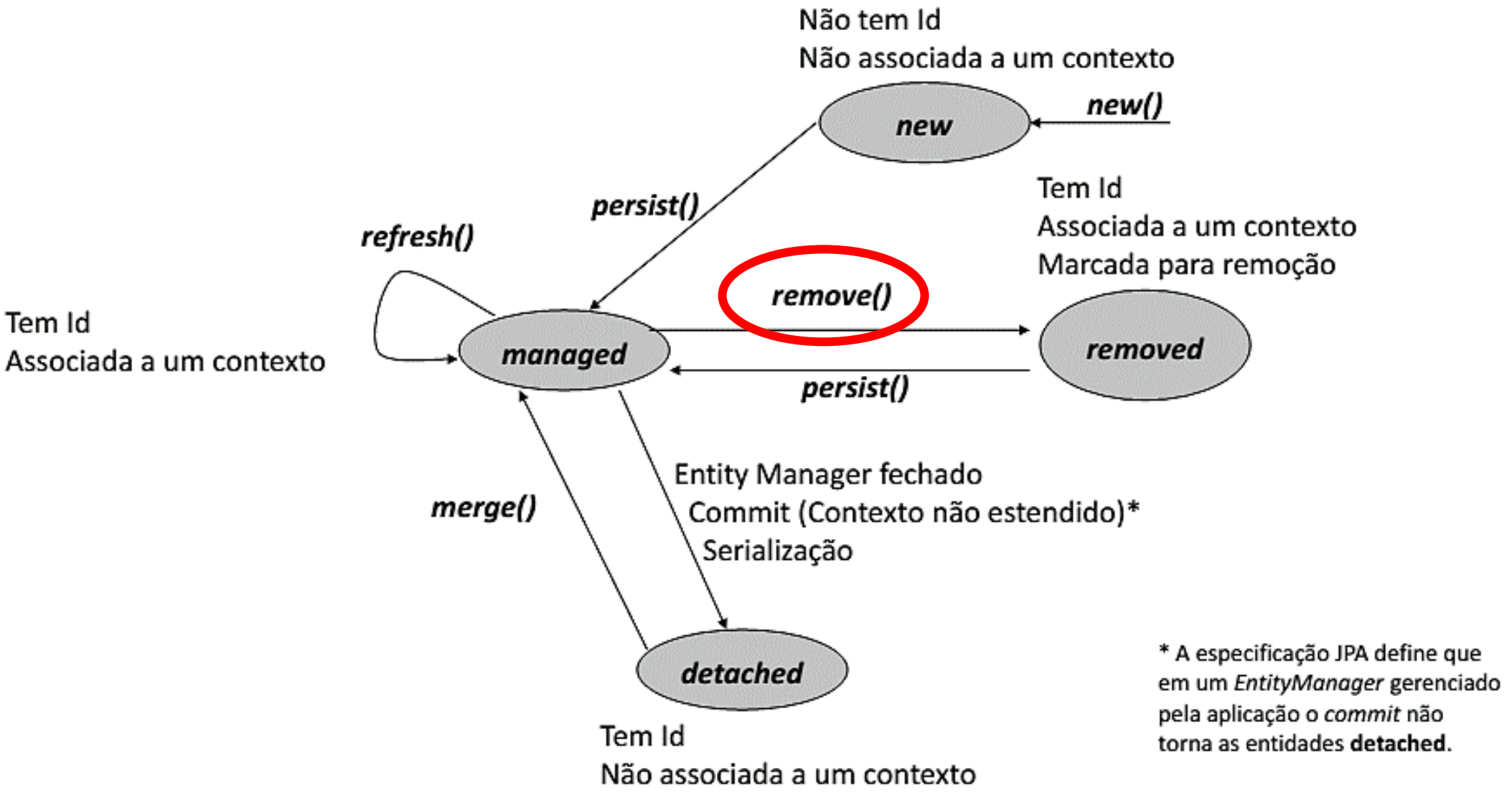
- **remove(Object entidade)**: marca uma entidade como **REMOVED** que será excluída do banco de dados após uma instrução **commit;**
  - ✓ Caso a entidade seja **NEW**, ela é ignorada;
  - ✓ Caso a entidade seja **MANAGED**, ela torna-se **REMOVED;**
  - ✓ Caso a entidade seja **REMOVED** ela é ignorada;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

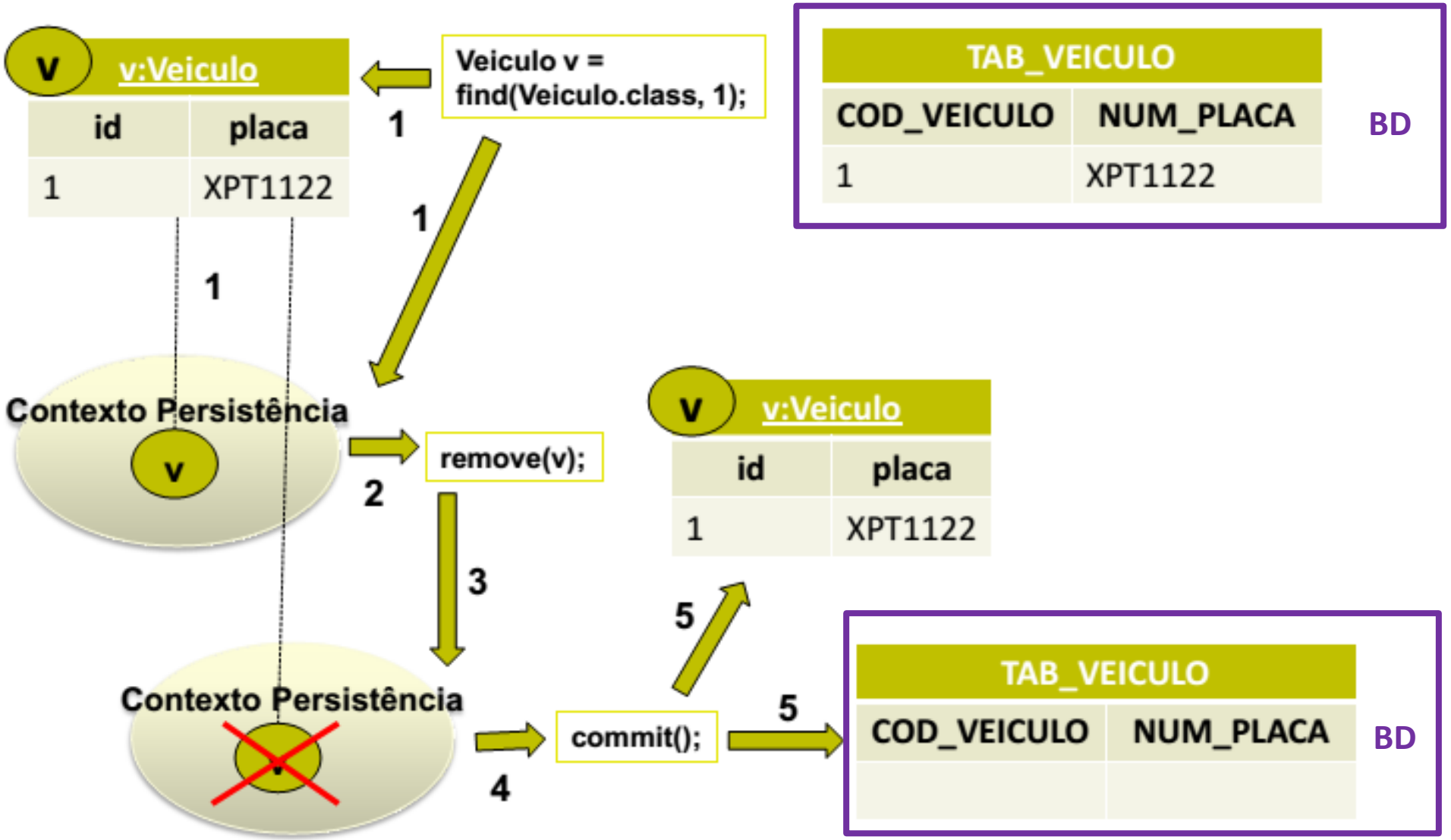
```
Veiculo veiculo = manager.find(Veiculo.class, 10);
manager.remove(veiculo);
```

ESTADOS DA ENTIDADE



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

REMOVE


v
v:Veiculo
id
placa
1
XPT1122
Veiculo v =
find(Veiculo.class, 1);
1
1
1
Contexto Persistência
v
remove(v);
2
3
Contexto Persistência
4
commit();
5
5
v
v:Veiculo
id
placa
1
XPT1122
TAB_VEICULO
COD_VEICULO
NUM_PLACA
1
XPT1122
BD
TAB_VEICULO
COD_VEICULO
NUM_PLACA
BD

MÉTODOS DO CICLO DE VIDA

- 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;

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

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