

<http://rogerioaraujo.wordpress.com>

**Série Rações Semanais**

# **JPA & Hibernate**

**Rogério Araújo**

<http://rogerioaraujo.wordpress.com>

**Série Rações Semanais**

# JPA & Hibernate

**Rogério Araújo**

## Questões

---

*A adversidade desperta em nós capacidades que, em circunstâncias favoráveis, teriam ficado adormecidas.*

Horácio

### FCC 2011 TRT 23ª Região – Técnico Judiciário – Tecnologia da Informação

42 Considere:

```
<hibernate-mapping>

    <class name="events.Event" table="EVENTS">
        <id name="id" column="EVENT_ID">
            <generator class="native"/>
        </id>
        <property name="date" type="timestamp" column="EVENT_DATE"/>
        <property name="title"/>
    </class>

</hibernate-mapping>
```

Em relação ao conteúdo do arquivo de mapeamento do Hibernate é INCORRETO afirmar que o

- A) elemento generator especifica a estratégia de geração do identificador, que neste caso, é native.
- B) atributo name do elemento property informa ao Hibernate qual método getter e setter deverá usar.
- C) Hibernate irá procurar pelo getDate()/setDate().
- D) Hibernate irá procurar pelo getTitle()/setTitle().
- E) mapeamento da propriedade title apresenta erro, pois omite os atributos type e column.

43 Considere:

```
@Entity
@Table(name = "domic")
@NamedQueries ({
    @NamedQuery(name = "Domic.findById", query = "SELECT r FROM Domic r WHERE r.id = :id"),
    @NamedQuery(name = "Domic.findByName", query = "SELECT r FROM Domic r WHERE r.nome = :nome")
})
public class Domic implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Column(name = "id", nullable = false)
    private Integer id;
    @Column(name = "nome")
    private String nome;
    @OneToMany(cascade = CascadeType.ALL, mappedBy = "domicId")
    private Collection<Predio> predioCollection;
```

Em relação à JPA (Java Persistence API) é INCORRETO afirmar que

- A) @NamedQuery é aplicada para definir várias consultas.

- B) @Entity define que haverá correspondência da classe com uma tabela do banco de dados.
- C) @Id define que o atributo que está mapeado com tal anotação corresponderá à chave primária da tabela.
- D) @Column(name = "id", nullable = false) define que o atributo da classe mapeado com tal anotação deve estar associado à coluna cujo nome é "id", além de definir que tal campo não pode ser nulo.
- E) @OneToMany indica que o atributo contém um conjunto de entidades que a referenciam.

## FCC 2011 TRT 23ª Região – Analista Judiciário – Tecnologia da Informação

**36** Sobre as coleções do Hibernate, considere:

- I. Quando um objeto é adicionado na coleção, o número da versão do proprietário da coleção é incrementado.
- II. Quando um objeto é removido da coleção, o número da versão do proprietário da coleção é decrementado.
- III. Se uma entidade é adicionada a uma coleção (uma associação um-para-muitos ou muitos-para-muitos), ela fará com que a entidade se torne persistente, por padrão.
- IV. Se um objeto removido de uma coleção for uma instância de um tipo de valor, este objeto irá parar de ser persistente e seu estado será completamente removido do banco de dados.

Está correto o que consta em

- a) I e III, apenas.
- b) I e IV, apenas.
- c) II e III, apenas.
- d) II e IV, apenas.
- e) I, II, III e IV.

## FCC 2011 TRT 14ª Região – Analista Judiciário – Tecnologia da Informação

**52** Quanto ao Hibernate no ambiente Java, considere:

- I. Sua principal característica é a transformação das classes Java para tabelas de dados porém não dos tipos de dados Java para os da SQL.
- II. Nas aplicações standalone, delega o tratamento transacional ao driver JDBC.
- III. Pode ser utilizado em aplicações Java standalone ou em aplicações Java EE.

É INCORRETO o que se afirma, APENAS em

- a) I.
- b) II.
- c) III.
- d) I e II.
- e) II e III.

## FCC 2011 TRE/RN – Técnico Judiciário – Tecnologia da Informação

**39** No Hibernate,

- A) os dados tabulares de um banco de dados são transformados em um grafo de objetos definido pelo desenvolvedor.
- B) a solução aos ambientes Java de mapeamento objeto/relacional é proporcionada pela abstração de toda a parte de representação de tipo de dados por meio de arquivos HTML.
- C) o desenvolvedor pode acessar diretamente uma tabela de banco de dados, bastando, para isso, abrir uma conexão.
- D) executa operações de busca, cadastro ou alteração de dados, sem requerer mapeamento da tabela em arquivo XML.
- E) INSERT é a única instrução SQL que requer utilização pelo desenvolvedor.

## FCC 2010 TRT 8ª Região – Analista Judiciário – Tecnologia da Informação

**43** Os três estados de objeto definidos pelo framework Hibernate são:

- A) Temporário (Temporary), Permanente (Permanent) e Resiliente (Resilient).
- B) Transiente (Transient), Persistente (Persistent) e Resiliente (Resilient).
- C) Temporário (Temporary), Persistente (Persistent) e Destacado (Detached).
- D) Transiente (Transient), Persistente (Persistent) e Destacado (Detached).
- E) Transiente (Transient), Permanente (Permanent) e Resiliente (Resilient).

## FCC 2010 TRT 22ª Região – Técnico Judiciário – Tecnologia da Informação

**47** Hibernate é um framework

- A) que separa as funções que envolvem a construção de aplicações Web, através da associação dos eventos do lado cliente com os manipuladores dos eventos do lado do servidor.
- B) pelo qual o programador utiliza a infraestrutura do servidor de aplicação voltada para o desenvolvimento de aplicações de missão crítica e de aplicações empresariais em geral.
- C) no qual as questões de infraestrutura, segurança, disponibilidade e escalabilidade são responsabilidade do servidor de aplicações, permitindo que o programador se concentre, apenas, nas necessidades do negócio do cliente.
- D) que permite ao desenvolvedor de páginas para internet produzir aplicações que acessem o banco de dados, manipulem arquivos no formato texto e capturem informações a partir de formulários.
- E) cujo objetivo é diminuir a complexidade entre os programas Java que precisam trabalhar com um banco de dados do modelo relacional.

## FCC 2010 MPE/SE – Analista do Ministério Público – Área Informática II – Gestão e Análise de Projeto de Sistema

**60** No framework Hibernate os objetos que ainda não têm uma representação no banco de dados, ainda não estão sob o controle do framework e podem não ser mais referenciáveis a qualquer momento, como qualquer objeto normal em Java, são do tipo

- A) detached.
- B) temporary.
- C) waiting.
- D) transient.
- E) featured.

## FCC 2009 TJ/PI – Analista Judiciário – Analista de Sistemas

**56** Hibernate é uma ferramenta open-source que facilita a comunicação entre aplicação Java e banco de dados, tais como manipulação de dados, gerenciamento de transações e pooling de conexões. A esse respeito, considere o trecho de código para configuração do Hibernate:

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
  <class name="Pessoa" table="PESSOA">
    <id name="idPessoa" column="ID_PESSOA">
      <generator class="increment"/>
    </id>
    <property name="nome" column="NOME_PESSOA"/>
    <property name="email"/>
  </class>
</hibernate-mapping>
```

Tendo em vista os dados acima, analise:

- I. o código exibe a estrutura básica de um arquivo XML de mapeamento que indica ao Hibernate como carregar e gravar os objetos de uma classe de persistência;
- II. as declarações de mapeamento estão circunscritas entre as tags </hibernate-mapping>;
- III. o elemento <id> é a declaração de uma propriedade identificadora de tabela (chave-primária) e o atributo column="ID\_PESSOA" indica qual é o campo na tabela ao qual ele está ligado.

É correto o que consta em

- A) I e II, apenas.
- B) I, II e III.
- C) II e III, apenas.
- D) I e III, apenas.
- E) I, apenas.

## FCC 2007 TRE/SE – Analista Judiciário – Analista de Sistemas

**42** Sendo um grupo de classes e componentes responsáveis pelo armazenamento e recuperação de dados, esta camada inclui necessariamente um modelo das entidades do domínio de negócios (mesmo que seja somente um modelo de metadados). No âmbito do mapeamento objeto-relacional (hibernate) esta é a camada de

- A) negócio.
- B) restrição.
- C) apresentação.
- D) consistência.
- E) persistência.

## FCC 2007 MPU – Analista – Área Informática – Desenvolvimento de Sistemas

**59** Objetos que têm uma representação no banco de dados, mas não fazem mais parte de uma sessão do Hibernate, o que significa que o seu estado pode não estar mais sincronizado com o banco de dados, são do tipo

- A) transient.
- B) detached.
- C) attached.
- D) persistent.
- E) consistent.

## CESPE 2011 CORREIOS – Cargo 3

**58** No Hibernate, o recurso Query Cache possibilita fazer o cache de queries que são executadas várias vezes.

## CESPE 2010 TRE/BA – Cargo 21

**54** As tecnologias JPA e EJB permitem, com o uso da linguagem Java, a manipulação de dados que estão em um banco de dados.

**107** O Hibernate, um framework para o mapeamento objeto-relacional, é escrito na linguagem Java e, por isso, somente pode ser executado no ambiente Java.

## CESPE 2010 TRE/BA – Cargo 3

**66** No Hibernate, apenas a linguagem de consulta HQL (hibernate query language) pode ser utilizada. A HQL executa os pedidos SQL sobre as classes de persistência do Java em vez de tabelas no banco de dados, o que diminui a distância entre o desenvolvimento das regras de negócio e o banco de dados.



## CESPE 2010 MPU – Cargo 26

**73** Na instalação padrão do Hibernate, o trecho de código Java abaixo permite a inserção de 200.000 linhas em uma tabela no banco de dados.

```
Session ses = sessionFactory.openSession();
Transaction trans = ses.beginTransaction();
for ( int i=0; i<200000; i++ ) {
    Pessoa p = new Pessoa();
    ses.save(p);
}
trans.commit();
ses.close();
```

**74** A conexão da aplicação com o banco de dados pode ser realizada por meio da rede de conexão interna do Hibernate, que é bastante eficiente para uso em ambiente de produção.

**75** O C3P0 é uma rede de conexão JDBC, de código aberto, distribuído junto com o Hibernate, no diretório lib.

## CESPE 2010 IJSN/ES

**71** O ciclo de vida do Hibernate consiste basicamente em três estados de persistência: transiente, persistente e desacoplado. Objetos transientes são todos aqueles que não possuem a anotação @Persistent. Uma vantagem de um objeto transiente é que ele conta com a possibilidade de rollback até que seja persistido utilizando o método save() ou o saveOrUpdate().

**72** O Hibernate permite diversas estratégias de fetching, como o join fetching, que permite escolher quais objetos relacionados devem ser carregados juntamente com o objeto referenciado. Dessa forma, esses objetos associados podem ser retornados em uma única solicitação ao banco de dados. Estratégias utilizando o join fetching podem ser aplicadas para se obter melhorias de desempenho no Hibernate.

## CESPE 2010 DETRAN/ES

**74** O Hibernate, framework utilizado no desenvolvimento de consultas e atualização de dados em um banco relacional, foi criado para facilitar a integração entre programas em Java, funcionando também em ambientes .Net (NHibernate).

## CESPE 2009 SECONT/ES

**88** O Hibernate, um framework de mapeamento objeto relacional (ORM), cria uma camada persistência na solução desenvolvida, o que permite ligar os objetos aos bancos de dados relacionais. Entre seus serviços, o Hibernate provê um meio de se controlar transações, por meio de métodos de suas interfaces session e transaction, tendo ainda suporte a herança e polimorfismo. É distribuído sob a licença LGPL, o que permite seu uso em projetos comerciais ou open source.



## CESPE 2009 CEHAP/PB

**29** No framework Hibernate, é comum que uma instância de uma classe persistente tenha três estados específicos. Assinale a opção que contém esses três estados.

- A) plugged, disconnected, timewait
- B) connected, disconnected, detached
- C) transient, persistent, detached
- D) transient, connected, timewait

## CESPE 2008 TJ/CE

```
1 <?xml version="1.0"?>
2 <!DOCTYPE hibernate-mapping PUBLIC
3     "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
4     "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
5
6 <hibernate-mapping>
7     <class name="events.Event" table="EVENTS">
8         <id name="id" column="EVENT_ID">
9             <generator class="native"/>
10        </id>
11    </class>
12 </hibernate-mapping>
```

Apresenta-se acima o conteúdo de um documento XML usado no framework hibernate. Julgue os seguintes itens relativos às informações apresentadas e aos conceitos de desenvolvimento de aplicações e bancos de dados.

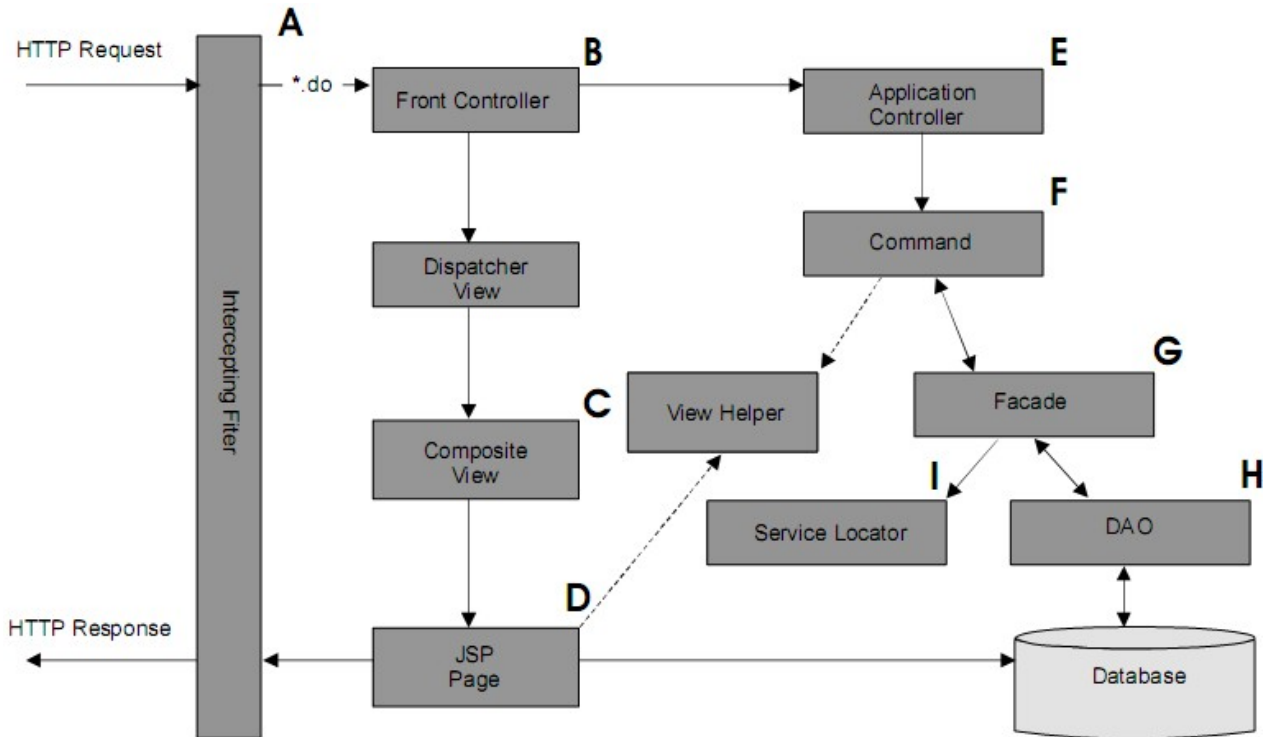
**78** A aplicação que utiliza esse mapeamento faz referência a uma classe chamada Event, que está declarada no package default.

**79** A aplicação que utiliza esse mapeamento depende de um banco de dados ou esquema que possui a tabela denominada Event e uma sequence. A referida tabela contém uma coluna denominada EVENT\_ID, que é sua chave primária.

**80** O documento declara apenas três elementos XML, sendo: um vazio; outro que não declara atributos; e outro que possui mais de um atributo declarado.

**81** O documento usa namespaces XML.

## CESPE 2008 STF – Cargo 3



A figura acima, adaptada de java.sun.com, ilustra a arquitetura de uma aplicação web desenvolvida na plataforma J2EE, tendo sido alguns de seus módulos nomeados de A até I. Considere que uma aplicação com a arquitetura mostrada tenha sido instalada em um servidor de aplicação JBoss 4.0 ou superior, por meio do deploy de um arquivo com nome aplicacao.war, e se encontre em pleno funcionamento. Com base nessas informações, julgue os itens de 105 a 109, quanto aos conceitos de desenvolvimento de aplicações.

**108** Considere que, no cenário da aplicação mostrada, o framework Hibernate, versão 3.0, esteja em pleno uso. Nesse caso, considerando-se a prática recomendada no desenvolvimento de aplicações web, é correto afirmar que os módulos G e I não deverão importar classes do package org.hibernate, mas que o módulo H deverá importar classes e interfaces do package org.hibernate e, possivelmente, do package java.sql.

## CESPE 2008 SERPRO – Cargo 5

**94** O objetivo do framework Hibernate é facilitar a integração entre os modelos orientados a objeto e os que usam banco de dados no modelo relacional.

## CESPE 2008 MPE/RR – Cargo 3

```

1 package events;
2 import org.hibernate.Session;
3 import java.util.Date;
4 import util.HibernateUtil;
5 public class EventManager {
6     public static void main(String[] args) {
7         EventManager mgr = new EventManager();
8         if (args[0].equals("store")) {
9             mgr.createAndStoreEvent("My Event", new Date());
10        }
11        HibernateUtil.getSessionFactory().close();
12    }
13    private void createAndStoreEvent(String title, Date theDate) {
14        Session session = HibernateUtil.getSessionFactory().getCurrentSession();
15        session.beginTransaction();
16        Event theEvent = new Event();
17        theEvent.setTitle(title);
18        theEvent.setDate(theDate);
19        session.save(theEvent);
20        session.getTransaction().commit();
21    }
22 }

```

Acima, apresenta-se um código sintática e semanticamente correto de uma aplicação Java, adaptado a partir de <http://www.hibernate.org>. A aplicação adota os padrões de configuração de persistência objeto-relacional propostos no framework Hibernate. Considerando essas informações e o caso de uma execução bem-sucedida dessa aplicação, julgue os itens seguintes, acerca de conceitos da linguagem e frameworks Java.

**116** As informações de configuração para permitir a conexão com o SGBD devem estar obrigatoriamente acessíveis através da classe HibernateUtil.

**117** No mesmo diretório onde se encontra a classe EventManager, é plausível a existência de um arquivo XML de nome Event.hbm.xml, onde são usadas as tags hibernate-mapping e class.

**118** A eliminação das linhas 15, 19 e 20 do código acima impedirá sua subsequente compilação.

## CESPE 2008 MPE/AM – Cargo 11

Julgue os itens a seguir, acerca da persistência transparente do Hibernate e do framework Hibernate.

**111** Nesse framework, o método de conexão é intrusivo porque requer interfaces ou base de classes.

**112** O Hibernate possui suporte a tipos definidos pelo usuário para os modelos de persistência.

**113** A aplicação deve se preocupar com violações de constraints de chaves estrangeiras.

**114** O Hibernate não usa geração de bytecode em tempo de execução porque não propaga códigos via banco relacional.

**115** Hibernate é desenvolvido nativamente em java e requer, no mínimo, a versão 1.4 ou superior.

## CESPE 2008 Ministério das Comunicações

**115** Acerca do framework Hibernate, é correto afirmar que ele: possibilita o mapeamento entre o modelo relacional e o orientado a objetos e pode ser definido usando-se arquivos XML; ainda que, em um arquivo de configuração, pode-se declarar uma classe persistente, usando o elemento “class”; além disso, os relacionamentos entre as classes persistentes têm que ser unidirecionais e 1:1.

**116** Quanto ao framework Hibernate é correto afirmar que: nas aplicações que o usam, as classes persistentes devem seguir o modelo JavaBeans, pois é impossível persistir as classes que seguem o modelo POJO (plain old java object); o framework só deve ser usado naquelas aplicações que usam intensamente procedimentos armazenados (stored procedures).

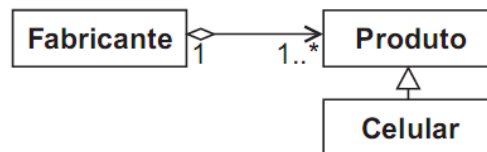
## CESPE 2008 Corpo de Bombeiros Militar/DF – Cargo 27

**107** Pode-se informar ao Hibernate como carregar e armazenar objetos de classes persistentes via tags em arquivos de mapeamento. Em um arquivo de mapeamento, tags <class> podem ser usadas para informar ao Hibernate os nomes das classes e das tabelas onde armazenar as instâncias.

**108** É possível informar ao Hibernate os identificadores que são chaves primárias e as colunas nas quais armazená-los. O Hibernate não possibilita associações entre classes persistentes cuja multiplicidade seja N: M. O Hibernate também não possibilita o armazenamento de associações bidirecionais.

## CESGRANRIO 2008 TJ/RO – Agente Judiciário – Alista de Sistemas (Microinformática)

26



O modelo mostrado no diagrama de classes acima foi mapeado para um esquema relacional em um banco de dados, e será utilizado em uma aplicação Java com Hibernate em sua configuração padrão. Todas as classes do modelo são concretas.

```

List x = session.createQuery(
    "from Produto as produto " +
    "where produto.fabricante.codigo = ?")
    .setString(0, codigo).list();
    
```

Assumindo que existem registros no banco de dados que satisfazem à cláusula where da query HQL mostrada acima, a variável x será populada com

- A) os objetos da classe Produto associados ao Fabricante cujo código corresponde ao valor armazenado na variável codigo.
- B) os objetos das classes Produto e Celular associados ao Fabricante cujo código corresponde ao valor armazenado na variável codigo.
- C) os objetos das classes Produto e Celular associados ao Fabricante cujo codigo é 0.
- D) tuplas (Object[]) cujo primeiro elemento é um Produto e o segundo elemento, seu Fabricante.
- E) null, pois nada foi selecionado na query, já que nenhuma cláusula select foi especificada.

## Gabarito

---

*É graça divina começar bem. Graça maior persistir na caminhada certa.  
Mas graça das graças é não desistir nunca.*

*Dom Hélder Câmara*

### FCC 2011 TRT 23ª Região – Técnico Judiciário – Tecnologia da Informação

**42** Considere:

```
<hibernate-mapping>

    <class name="events.Event" table="EVENTS">
        <id name="id" column="EVENT_ID">
            <generator class="native"/>
        </id>
        <property name="date" type="timestamp" column="EVENT_DATE"/>
        <property name="title"/>
    </class>

</hibernate-mapping>
```

Em relação ao conteúdo do arquivo de mapeamento do Hibernate é INCORRETO afirmar que o

- a) elemento generator especifica a estratégia de geração do identificador, que neste caso, é native.
- b) atributo name do elemento property informa ao Hibernate qual método getter e setter deverá usar.
- c) Hibernate irá procurar pelo getDate()/setDate().
- d) Hibernate irá procurar pelo getTitle()/setTitle().
- e) mapeamento da propriedade title apresenta erro, pois omite os atributos type e column.

**Gabarito: letra E.**

**43** Considere:

```
@Entity
@Table(name = "domic")
@NamedQueries ({
    @NamedQuery(name = "Domic.findById", query = "SELECT r FROM Domic r WHERE r.id = :id"),
    @NamedQuery(name = "Domic.findByName", query = "SELECT r FROM Domic r WHERE r.nome = :nome")
})
public class Domic implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Column(name = "id", nullable = false)
    private Integer id;
    @Column(name = "nome")
    private String nome;
    @OneToMany(cascade = CascadeType.ALL, mappedBy = "domicId")
    private Collection<Predio> predioCollection;
```

Em relação à JPA (Java Persistence API) é INCORRETO afirmar que

- A) @NamedQuery é aplicada para definir várias consultas.
- B) @Entity define que haverá correspondência da classe com uma tabela do banco de dados.
- C) @Id define que o atributo que está mapeado com tal anotação corresponderá à chave primária da tabela.
- D) @Column(name = "id", nullable = false) define que o atributo da classe mapeado com tal anotação deve estar associado à coluna cujo nome é "id", além de definir que tal campo não pode ser nulo.
- E) @OneToMany indica que o atributo contém um conjunto de entidades que a referenciam.

**Gabarito: letra A.**

## FCC 2011 TRT 23ª Região – Analista Judiciário – Tecnologia da Informação

**36** Sobre as coleções do Hibernate, considere:

- I. Quando um objeto é adicionado na coleção, o número da versão do proprietário da coleção é incrementado.
- II. Quando um objeto é removido da coleção, o número da versão do proprietário da coleção é decrementado.
- III. Se uma entidade é adicionada a uma coleção (uma associação um-para-muitos ou muitos-para-muitos), ela fará com que a entidade se torne persistente, por padrão.
- IV. Se um objeto removido de uma coleção for uma instância de um tipo de valor, este objeto irá parar de ser persistente e seu estado será completamente removido do banco de dados.

Está correto o que consta em

- a) I e III, apenas.
- b) I e IV, apenas.
- c) II e III, apenas.
- d) II e IV, apenas.
- e) I, II, III e IV.

**Gabarito: letra B.**

## FCC 2011 TRT 14ª Região – Analista Judiciário – Tecnologia da Informação

**52** Quanto ao Hibernate no ambiente Java, considere:

- I. Sua principal característica é a transformação das classes Java para tabelas de dados porém não dos tipos de dados Java para os da SQL.
- II. Nas aplicações standalone, delega o tratamento transacional ao driver JDBC.
- III. Pode ser utilizado em aplicações Java standalone ou em aplicações Java EE.

É INCORRETO o que se afirma, APENAS em

- a) I.
- b) II.

- c) III.
- d) I e II.
- e) II e III.

**Gabarito: letra A.**

## FCC 2011 TRE/RN – Técnico Judiciário – Tecnologia da Informação

**39** No Hibernate,

- A) os dados tabulares de um banco de dados são transformados em um grafo de objetos definido pelo desenvolvedor.
- B) a solução aos ambientes Java de mapeamento objeto/relacional é proporcionada pela abstração de toda a parte de representação de tipo de dados por meio de arquivos HTML.
- C) o desenvolvedor pode acessar diretamente uma tabela de banco de dados, bastando, para isso, abrir uma conexão.
- D) executa operações de busca, cadastro ou alteração de dados, sem requerer mapeamento da tabela em arquivo XML.
- E) INSERT é a única instrução SQL que requer utilização pelo desenvolvedor.

**Gabarito: letra A.**

## FCC 2010 TRT 8ª Região – Analista Judiciário – Tecnologia da Informação

**43** Os três estados de objeto definidos pelo framework Hibernate são:

- A) Temporário (Temporary), Permanente (Permanent) e Resiliente (Resilient).
- B) Transiente (Transient), Persistente (Persistent) e Resiliente (Resilient).
- C) Temporário (Temporary), Persistente (Persistent) e Destacado (Detached).
- D) Transiente (Transient), Persistente (Persistent) e Destacado (Detached).
- E) Transiente (Transient), Permanente (Permanent) e Resiliente (Resilient).

**Gabarito: letra D.**

## FCC 2010 TRT 22ª Região – Técnico Judiciário – Tecnologia da Informação

**47** Hibernate é um framework

- A) que separa as funções que envolvem a construção de aplicações Web, através da associação dos eventos do lado cliente com os manipuladores dos eventos do lado do servidor.
- B) pelo qual o programador utiliza a infraestrutura do servidor de aplicação voltada para o desenvolvimento de aplicações de missão crítica e de aplicações empresariais em geral.
- C) no qual as questões de infraestrutura, segurança, disponibilidade e escalabilidade são responsabilidade do servidor de aplicações, permitindo que o programador se concentre, apenas, nas necessidades do negócio do cliente.



- D) que permite ao desenvolvedor de páginas para internet produzir aplicações que acessem o banco de dados, manipulem arquivos no formato texto e capturem informações a partir de formulários.
- E) cujo objetivo é diminuir a complexidade entre os programas Java que precisam trabalhar com um banco de dados do modelo relacional.

**Gabarito: letra E.**

## FCC 2010 MPE/SE – Analista do Ministério Público – Área Informática II – Gestão e Análise de Projeto de Sistema

**60** No framework Hibernate os objetos que ainda não têm uma representação no banco de dados, ainda não estão sob o controle do framework e podem não ser mais referenciáveis a qualquer momento, como qualquer objeto normal em Java, são do tipo

- A) detached.
- B) temporary.
- C) waiting.
- D) transient.
- E) featured.

**Gabarito: letra D.**

## FCC 2009 TJ/PI – Analista Judiciário – Analista de Sistemas

**56** Hibernate é uma ferramenta open-source que facilita a comunicação entre aplicação Java e banco de dados, tais como manipulação de dados, gerenciamento de transações e pooling de conexões. A esse respeito, considere o trecho de código para configuração do Hibernate:

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
  <class name="Pessoa" table="PESSOA">
    <id name="idPessoa" column="ID_PESSOA">
      <generator class="increment"/>
    </id>
    <property name="nome" column="NOME_PESSOA"/>
    <property name="email"/>
  </class>
</hibernate-mapping>
```

Tendo em vista os dados acima, analise:

- I. o código exibe a estrutura básica de um arquivo XML de mapeamento que indica ao Hibernate como carregar e gravar os objetos de uma classe de persistência;
- II. as declarações de mapeamento estão circunscritas entre as tags </hibernate-mapping>;
- III. o elemento <id> é a declaração de uma propriedade identificadora de tabela (chave-primária) e o atributo column="ID\_PESSOA" indica qual é o campo na tabela ao qual ele está ligado.

É correto o que consta em

- A) I e II, apenas.

- B) I, II e III.
- C) II e III, apenas.
- D) I e III, apenas.
- E) I, apenas.

**Gabarito: letra B.**

## FCC 2007 TRE/SE – Analista Judiciário – Analista de Sistemas

**42** Sendo um grupo de classes e componentes responsáveis pelo armazenamento e recuperação de dados, esta camada inclui necessariamente um modelo das entidades do domínio de negócios (mesmo que seja somente um modelo de metadados). No âmbito do mapeamento objeto-relacional (hibernate) esta é a camada de

- A) negócio.
- B) restrição.
- C) apresentação.
- D) consistência.
- E) persistência.

**Gabarito: letra E.**

## FCC 2007 MPU – Analista – Área Informática – Desenvolvimento de Sistemas

**59** Sendo um grupo de classes e componentes responsáveis pelo armazenamento e recuperação de dados, esta camada inclui necessariamente um modelo das entidades do domínio de negócios (mesmo que seja somente um modelo de metadados). No âmbito do mapeamento objeto-relacional (hibernate) esta é a camada de

- A) transient.
- B) detached.
- C) attached.
- D) persistent.
- E) consistent.

**Gabarito: letra B.**

## CESPE 2011 CORREIOS – Cargo 3

**58** No Hibernate, o recurso Query Cache possibilita fazer o cache de queries que são executadas várias vezes.

**Gabarito: CERTO.**

## CESPE 2010 TRE/BA – Cargo 21

**54** As tecnologias JPA e EJB permitem, com o uso da linguagem Java, a manipulação de dados que estão em um banco de dados.

Gabarito: **CERTO**.

**107** O Hibernate, um framework para o mapeamento objeto-relacional, é escrito na linguagem Java e, por isso, somente pode ser executado no ambiente Java.

Gabarito: **ERRADO**.

## CESPE 2010 TRE/BA – Cargo 3

**66** No Hibernate, apenas a linguagem de consulta HQL (hibernate query language) pode ser utilizada. A HQL executa os pedidos SQL sobre as classes de persistência do Java em vez de tabelas no banco de dados, o que diminui a distância entre o desenvolvimento das regras de negócio e o banco de dados.

Gabarito: **ERRADO**.

## CESPE 2010 MPU – Cargo 26

**73** Na instalação padrão do Hibernate, o trecho de código Java abaixo permite a inserção de 200.000 linhas em uma tabela no banco de dados.

```
Session ses = sessionFactory.openSession();
Transaction trans = ses.beginTransaction();
for ( int i=0; i<200000; i++ ) {
    Pessoa p = new Pessoa();
    ses.save(p);
}
trans.commit();
ses.close();
```

Gabarito: **ERRADO**.

**74** A conexão da aplicação com o banco de dados pode ser realizada por meio da rede de conexão interna do Hibernate, que é bastante eficiente para uso em ambiente de produção.

Gabarito: **ERRADO**.

**75** O C3P0 é uma rede de conexão JDBC, de código aberto, distribuído junto com o Hibernate, no diretório lib.

Gabarito: **CERTO**.

## CESPE 2010 IJSN/ES

**71** O ciclo de vida do Hibernate consiste basicamente em três estados de persistência: transiente, persistente e desacoplado. Objetos transientes são todos aqueles que não possuem a anotação

@Persistent. Uma vantagem de um objeto transiente é que ele conta com a possibilidade de rollback até que seja persistido utilizando o método save() ou o saveOrUpdate().

**Gabarito:** **ERRADO**.

**72** O Hibernate permite diversas estratégias de fetching, como o join fetching, que permite escolher quais objetos relacionados devem ser carregados juntamente com o objeto referenciado. Dessa forma, esses objetos associados podem ser retornados em uma única solicitação ao banco de dados. Estratégias utilizando o join fetching podem ser aplicadas para se obter melhorias de desempenho no Hibernate.

**Gabarito:** **CERTO**.

## CESPE 2010 DETRAN/ES

**74** O Hibernate, framework utilizado no desenvolvimento de consultas e atualização de dados em um banco relacional, foi criado para facilitar a integração entre programas em Java, funcionando também em ambientes .Net (NHibernate).

**Gabarito:** **CERTO**.

## CESPE 2009 SECONT/ES

**88** O Hibernate, um framework de mapeamento objeto relacional (ORM), cria uma camada persistência na solução desenvolvida, o que permite ligar os objetos aos bancos de dados relacionais. Entre seus serviços, o Hibernate provê um meio de se controlar transações, por meio de métodos de suas interfaces session e transaction, tendo ainda suporte a herança e polimorfismo. É distribuído sob a licença LGPL, o que permite seu uso em projetos comerciais ou open source.

**Gabarito:** **CERTO**.

## CESPE 2009 INMETRO – Cargo 5

**89** Considerando que para o uso da tecnologia Hibernate na linguagem Java são empregados, usualmente, dois tipos de arquivos: (i) configuração e (ii) mapeamento, sendo a lista a seguir uma sequência de elementos XML utilizados na tecnologia Hibernate: (a) hibernate-mapping (b) class; (c) generator; (d) property; (e) session-factory; e (f) mapping, então, uma associação adequada entre os elementos e o tipo de arquivo Hibernate, no qual eles usualmente são empregados, é a seguinte: {(a, ii), (b, ii), (c, i), (d, ii), (d, i), (e, i), (f, i)}.

**Gabarito:** **ERRADO**.

## CESPE 2009 CEHAP/PB

**29** No framework Hibernate, é comum que uma instância de uma classe persistente tenha três estados específicos. Assinale a opção que contém esses três estados.

- A) plugged, disconnected, timewait
- B) connected, disconnected, detached
- C) transient, persistent, detached

D) transient, connected, timewait

**Gabarito:** letra C.

## CESPE 2008 TJ/CE

```
1 <?xml version="1.0"?>
2 <!DOCTYPE hibernate-mapping PUBLIC
3     "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
4     "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
5
6 <hibernate-mapping>
7     <class name="events.Event" table="EVENTS">
8         <id name="id" column="EVENT_ID">
9             <generator class="native"/>
10        </id>
11    </class>
12 </hibernate-mapping>
```

Apresenta-se acima o conteúdo de um documento XML usado no framework hibernate. Julgue os seguintes itens relativos às informações apresentadas e aos conceitos de desenvolvimento de aplicações e bancos de dados.

**78** A aplicação que utiliza esse mapeamento faz referência a uma classe chamada Event, que está declarada no package default.

**Gabarito:** **ERRADO**.

**79** A aplicação que utiliza esse mapeamento depende de um banco de dados ou esquema que possui a tabela denominada Event e uma sequence. A referida tabela contém uma coluna denominada EVENT\_ID, que é sua chave primária.

**Gabarito:** **ERRADO**.

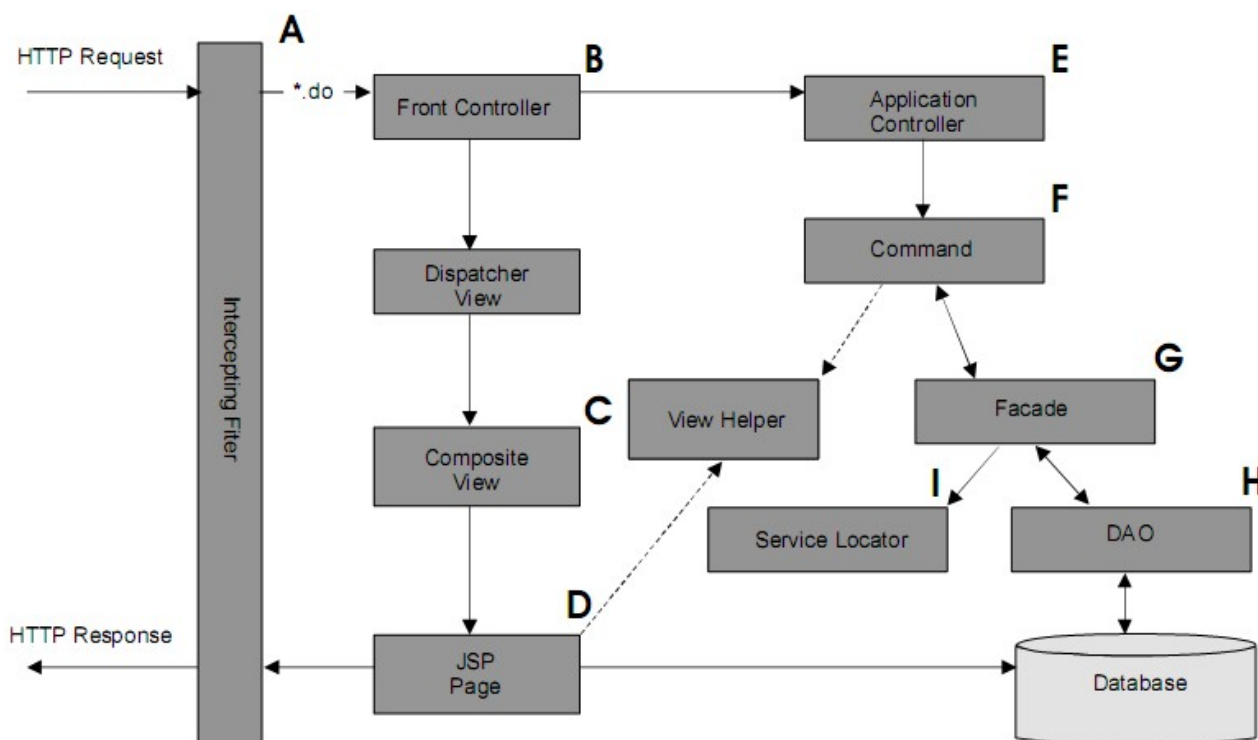
**80** O documento declara apenas três elementos XML, sendo: um vazio; outro que não declara atributos; e outro que possui mais de um atributo declarado.

**Gabarito:** **ERRADO**.

**81** O documento usa namespaces XML.

**Gabarito:** **ERRADO**.

## CESPE 2008 STF – Cargo 3



A figura acima, adaptada de java.sun.com, ilustra a arquitetura de uma aplicação web desenvolvida na plataforma J2EE, tendo sido alguns de seus módulos nomeados de A até I. Considere que uma aplicação com a arquitetura mostrada tenha sido instalada em um servidor de aplicação JBoss 4.0 ou superior, por meio do deploy de um arquivo com nome aplicacao.war, e se encontre em pleno funcionamento. Com base nessas informações, julgue os itens de 105 a 109, quanto aos conceitos de desenvolvimento de aplicações.

**108** Considere que, no cenário da aplicação mostrada, o framework Hibernate, versão 3.0, esteja em pleno uso. Nesse caso, considerando-se a prática recomendada no desenvolvimento de aplicações web, é correto afirmar que os módulos G e I não deverão importar classes do package org.hibernate, mas que o módulo H deverá importar classes e interfaces do package org.hibernate e, possivelmente, do package java.sql.

**Gabarito:** CERTO.

## CESPE 2008 SERPRO – Cargo 5

**94** O objetivo do framework Hibernate é facilitar a integração entre os modelos orientados a objeto e os que usam banco de dados no modelo relacional.

**Gabarito:** ANULADO.

**Justificativa:** anulado porque a redação do item é ambígua, o que prejudica o seu julgamento. Dessa forma, o CESPE/UnB decide por sua anulação.

## CESPE 2008 MPE/RR – Cargo 3

```

1 package events;
2 import org.hibernate.Session;
3 import java.util.Date;
4 import util.HibernateUtil;
5 public class EventManager {
6     public static void main(String[] args) {
7         EventManager mgr = new EventManager();
8         if (args[0].equals("store")) {
9             mgr.createAndStoreEvent("My Event", new Date());
10        }
11        HibernateUtil.getSessionFactory().close();
12    }
13    private void createAndStoreEvent(String title, Date theDate) {
14        Session session = HibernateUtil.getSessionFactory().getCurrentSession();
15        session.beginTransaction();
16        Event theEvent = new Event();
17        theEvent.setTitle(title);
18        theEvent.setDate(theDate);
19        session.save(theEvent);
20        session.getTransaction().commit();
21    }
22 }

```

Acima, apresenta-se um código sintática e semanticamente correto de uma aplicação Java, adaptado a partir de <http://www.hibernate.org>. A aplicação adota os padrões de configuração de persistência objeto-relacional propostos no framework Hibernate. Considerando essas informações e o caso de uma execução bem-sucedida dessa aplicação, julgue os itens seguintes, acerca de conceitos da linguagem e frameworks Java.

**116** As informações de configuração para permitir a conexão com o SGBD devem estar obrigatoriamente acessíveis através da classe `HibernateUtil`.

**Gabarito:** CERTO.

**117** No mesmo diretório onde se encontra a classe `EventManager`, é plausível a existência de um arquivo XML de nome `Event.hbm.xml`, onde são usadas as tags `hibernate-mapping` e `class`.

**Gabarito:** CERTO.

**118** A eliminação das linhas 15, 19 e 20 do código acima impedirá sua subsequente compilação.

**Gabarito:** ERRADO.

## CESPE 2008 MPE/AM – Cargo 11

Julgue os itens a seguir, acerca da persistência transparente do Hibernate e do framework Hibernate.

**111** Nesse framework, o método de conexão é intrusivo porque requer interfaces ou base de classes.

**Gabarito:** ERRADO.

**112** O Hibernate possui suporte a tipos definidos pelo usuário para os modelos de persistência.

**Gabarito:** CERTO.

**113** A aplicação deve se preocupar com violações de constraints de chaves estrangeiras.

**Gabarito:** ERRADO.



**114** O Hibernate não usa geração de bytecode em tempo de execução porque não propaga códigos via banco relacional.

Gabarito: **ERRADO**.

**115** Hibernate é desenvolvido nativamente em java e requer, no mínimo, a versão 1.4 ou superior.

Gabarito: **CERTO**.

## CESPE 2008 Ministério das Comunicações

**115** Acerca do framework Hibernate, é correto afirmar que ele: possibilita o mapeamento entre o modelo relacional e o orientado a objetos e pode ser definido usando-se arquivos XML; ainda que, em um arquivo de configuração, pode-se declarar uma classe persistente, usando o elemento “class”; além disso, os relacionamentos entre as classes persistentes têm que ser unidirecionais e 1:1.

Gabarito: **ERRADO**.

**116** Quanto ao framework Hibernate é correto afirmar que: nas aplicações que o usam, as classes persistentes devem seguir o modelo JavaBeans, pois é impossível persistir as classes que seguem o modelo POJO (plain old java object); o framework só deve ser usado naquelas aplicações que usam intensamente procedimentos armazenados (stored procedures).

Gabarito: **ERRADO**.

## CESPE 2008 Corpo de Bombeiros Militar/DF – Cargo 27

**107** Pode-se informar ao Hibernate como carregar e armazenar objetos de classes persistentes via tags em arquivos de mapeamento. Em um arquivo de mapeamento, tags <class> podem ser usadas para informar ao Hibernate os nomes das classes e das tabelas onde armazenar as instâncias.

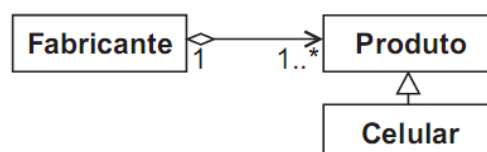
Gabarito: **CERTO**.

**108** É possível informar ao Hibernate os identificadores que são chaves primárias e as colunas nas quais armazená-los. O Hibernate não possibilita associações entre classes persistentes cuja multiplicidade seja N: M. O Hibernate também não possibilita o armazenamento de associações bidirecionais.

Gabarito: **ERRADO**.

## CESGRANRIO 2008 TJ/RO – Agente Judiciário – Alista de Sistemas (Microinformática)

26



O modelo mostrado no diagrama de classes acima foi mapeado para um esquema relacional em um banco de dados, e será utilizado em uma aplicação Java com Hibernate em sua configuração padrão. Todas as classes do modelo são concretas.

```
List x = session.createQuery(
    "from Produto as produto " +
    "where produto.fabricante.codigo = ?")
    .setString(0, codigo).list();
```

Assumindo que existem registros no banco de dados que satisfazem à cláusula where da query HQL mostrada acima, a variável x será populada com

- A) os objetos da classe Produto associados ao Fabricante cujo código corresponde ao valor armazenado na variável codigo.
- B) os objetos das classes Produto e Celular associados ao Fabricante cujo código corresponde ao valor armazenado na variável codigo.
- C) os objetos das classes Produto e Celular associados ao Fabricante cujo codigo é 0.
- D) tuplas (Object[]) cujo primeiro elemento é um Produto e o segundo elemento, seu Fabricante.
- E) null, pois nada foi selecionado na query, já que nenhuma cláusula select foi especificada.

**Gabarito: letra B.**

<http://rogerioaraujo.wordpress.com>

