**Respostas Fase 2**

I. Qual o papel do Driver JDBC no código acima?

*Fazer conexão e enviar instruções SQL ao banco de dados.*

II. O que é Statement?

*É usado para executar os códigos no banco de dados.*

III. Qual a diferença entre a chamada execute() e executeQuery() no JDBC?

*execute – executa qualquer tipo de operação.*

*executeQuery – retorna os dados do banco de dados.*

IV. O que é um ResultSet?

*É uma tabela de dados que representa um conjunto de dados do banco de dados.*

V. O que faz o método next() do ResultSet?

*Retorna um boolean. True ou False.*

VI. Como se obtêm o valor de uma coluna de um registro do banco de dados com JDBC?

Usa-se a função get. Exemplo: getInt ou getString.

**Respostas Fase 3**

I. O que é Unidade de Persistência?

*É usada para configurar as informações do banco de dados.*

II. Qual a relação do arquivo persistence.xml gerado na Fase 4 com a conexão JDBC criada na Fase 1?

*Não há relação.*

III. Onde no arquivo persistence.xml podemos mudar o nome de uma Unidade de Persistência?

*Na tag <persistence-unit name= “nome da unidade”>*

IV. Se quiséssemos conectar nossa aplicação com 2 bancos de dados distintos, o que poderíamos fazer?

*Inserir uma nova persistence-unit com um novo name.*

**Respostas Fase 4**

I. O que é um Entity Manager? Como o EntityManager foi criado? Que informação precisou ser informada na sua criação?

*É o serviço central para todas as ações de persistência, é ele que faz o gerenciamento. Foi criado “EntityManager em = emf.createEntityManager”, neste momento é criado a conexão com o banco de dados, aguardando apenas a abertura da transação e o comando “begin”.*

II. O que é uma transação de banco de dados? Quais linhas de código (duas) foram responsáveis por iniciar e finalizar a transação?

*É um conjunto de procedimentos que é executado num banco de dados.*

*em.getTransaction().begin(); (Inicializa a transação)*

*em.getTransaction().commit(); (Finaliza a transação e executa os dados)*

*em.getTransaction().rollback(); (Finalização e reverte ao passo anterior)*

III. Que método foi chamado para persistir uma entidade no banco de dados?

*Utilizando o método “EntityManagerFactory”, instanciamos uma persistência.*

IV. Que método foi chamado para listar entidades do banco de dados? O que foi informado para esse método?

*O método “createQuery” para criar a conexão. O parâmetro "from Carro" para que selecionasse todos os registros da tabela “Carro”.*

V. Quem criou o banco de dados?

*A tecnologia de persistência utiliza a propriedade de “Entity” nas classes para fazer um mapeamento idêntico nas classes.*

VI. Compare as linhas de código utilizadas para listar os Carros através do JPA com as linhas de código utilizadas para listar as Cores através do JDBC. Que facilidades foram trazidas pelo JPA?

*ResultSet rs = statement.executeQuery("SELECT \* FROM cores");*

*while (rs.next()) {*

*int id = rs.getInt("id");*

*String nome = rs.getString("descricao");*

*System.out.println("id=" + id + "; nome=" + nome);*

*}*

*List<Carro> carros = em.createQuery("from Carro").getResultList();*

*for (Carro c : carros) {*

*System.out.println(c.getPlaca());*

*}*

*Para listar no JPA é usado o método createQuery para buscar o dados do banco passando o parâmetro "from Carro" para que saiba em qual tabela buscar. O método getResultList() executa e busca os dados como uma lista onde é inserido numa lista de carro, depois só percorrer a lista listando os dados escolhidos.*

*Já no método JDBC precisamos executar a Query completa e criar variáveis idênticas à coluna que desejamos visualizar os dados. Para isso precisamos conhecer a tabela em toda sua estrutura.*

*Houve uma economia de linhas.*