

Aula 10

código 1

```
package aula10;

import static javax.persistence.GenerationType.SEQUENCE;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.SequenceGenerator;

@Entity
public class Contato {

    @Id
    @GeneratedValue(strategy = SEQUENCE, generator = "geradorPessoa")
    @SequenceGenerator(name = "geradorPessoa", sequenceName = "gerador_pessoa")
    private int codigo;

    private String nome;

    private String telefone;

    private String endereco;

    //getters e setters omitidos
}
```

```
package aula10;
import java.util.List;
import java.util.Scanner;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.EntityTransaction;
import javax.persistence.Persistence;
import javax.persistence.Query;

public class TesteContatoJpa {

    public static void main(String[] args) {

        EntityManagerFactory emf =
Persistence.createEntityManagerFactory("exemplo");
        EntityManager em = emf.createEntityManager();
        EntityTransaction tx = em.getTransaction();

        //cria um contato e ajusta seus valores
        Contato contato = new Contato();
```

```

Scanner scan = new Scanner(System.in);
System.out.print("\nDigite o nome:");
contato.setNome(scan.nextLine());
System.out.print("\nDigite o endereço:");
contato.setEndereco(scan.nextLine());
System.out.print("\nDigite o telefone:");
contato.setTelefone(scan.nextLine());

//adiciona um contato no banco
tx.begin();
em.persist(contato);
tx.commit();

//listar os contatos
Query query = em.createQuery("select e from Contato e");
List<Contato> listaContatos = query.getResultList();
for (Contato c : listaContatos) {
    System.out.printf("%20s %20s %20s", c.getNome(), c.getEndereco(),
c.getTelefone());
}
}
}

```

//Sintaxe básica

//SELECT objeto FROM entidade objeto

```

//Persistence.xml
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="1.0"
    xmlns="http://java.sun.com/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd">
    <persistence-unit name="exemplo">
        <class>aula10.Contato</class>
        <properties>
            <property name="hibernate.hbm2ddl.auto" value="update" />
            <property name="hibernate.dialect"
                value="org.hibernate.dialect.PostgreSQLDialect" />
            <property name="hibernate.max_fetch_depth" value="3" />
            <!-- mostra o sql no log -->
            <property name="hibernate.show_sql" value="true" />

            <!-- dados da conexao com o banco -->
            <property name="hibernate.connection.driver_class"
                value="org.postgresql.Driver" />
            <property name="hibernate.connection.username"
                value="postgres" />
            <property name="hibernate.connection.password"
                value="123456" />
            <property name="hibernate.connection.url"
                value="jdbc:postgresql://127.0.0.1/treinamento" />
        </properties>
    </persistence-unit>
</persistence>

```

código 2

```
package aula09;
//Observe que o pacote é da aula 09. não da aula11
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

public class ConexaoJpa {

    private static EntityManagerFactory emf;

    private static EntityManagerFactory
        criarEntityManagerFactory() {
        return
            Persistence.createEntityManagerFactory("agenda");
    }

    public static EntityManager getEntityManager() {
        if(emf == null){
            emf = criarEntityManagerFactory();
        }
        return emf.createEntityManager();
    }
}
```

```
package aula09;

import java.sql.Connection;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.EntityTransaction;
import javax.persistence.Persistence;

/**
 * Classe v.O. (Value Object), essa classe é responsável
 * por comunicar com a classe que acessa o banco,
 * e repassar o(s) objeto(s)
 * @author Edjalma
 */
```

```

* @method inserir(), alterar(), inserir(contatos), listar
*
*/

public class ContatoDao {
/**
 * Metodo responsavel por inserir um objeto
 * @param contato
 * @throws SQLException
 */
    public void inserir(Contato contato) throws SQLException {

        EntityManagerFactory emf =
            Persistence.createEntityManagerFactory("agenda");
        EntityManager em = emf.createEntityManager();
        EntityTransaction tx = em.getTransaction();

        try {
            tx.begin();
            em.persist(contato);
            tx.commit();
        } catch (Exception e) {

            e.printStackTrace();
        }
    }
/**
 * Metodo responsavel por alterar um objeto
 * @param contato
 */
    public void alterar(Contato contato) {
        EntityManagerFactory emf =
            Persistence.createEntityManagerFactory("agenda");
        EntityManager em = emf.createEntityManager();
        EntityTransaction tx = em.getTransaction();

        try {
            tx.begin();
            em.merge(contato);
            tx.commit();
        } catch (Exception e) {

            e.printStackTrace();
        }
    }

/**
 * Metodo responsavel por listar objetos
 * @return resultado
 */
    public List<Contato> listar() {

```

```

        Connection con = Conexao.getConexao();
        List<Contato> resultado = new ArrayList<Contato>();
        try {
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("select * from
contato order by nome");
            while (rs.next()) {
                Contato contato = new Contato();
                contato.setCodigo(rs.getInt("codigo"));
                contato.setNome(rs.getString("nome"));
                contato.setTelefone(rs.getString("telefone"));
                contato.setEndereco(rs.getString("endereco"));

                contato.setNascimento(rs.getDate("nascimento"));
                resultado.add(contato);
            }
            con.close();
        } catch (SQLException e) {
            System.out.println("Não foi possível consultar os
registros . . .");
        }
        return resultado;
    }
}

```

```
package aula11;
```

```
import static javax.persistence.GenerationType.SEQUENCE;
```

```
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.SequenceGenerator;
```

```
@Entity
```

```
public class Contato {
```

```
    @Id
```

```
    @GeneratedValue(strategy = SEQUENCE, generator =
"geradorPessoa")
```

```
    @SequenceGenerator(name = "geradorPessoa",
        sequenceName = "gerador_pessoa",
        initialValue=1, allocationSize=1)
```

```
    private int codigo;
```

```
    private String nome;
```

```
    private String telefone;
```

```
    private String endereco;
```

```
//getters and setters omitidos
```

```
}
```

```
package aula09;
```

```
import java.util.List;
```

```
import javax.persistence.EntityManager;
```

```
import javax.persistence.EntityTransaction;
```

```
import javax.persistence.Query;
```

```
public class ContatoDaoJpa {
```

```
    public void incluir(Contato contato){
        EntityManager em = ConexaoJpa.getEntityManager();
        EntityTransaction tx = em.getTransaction();
        tx.begin();
        em.persist(contato);
        tx.commit();
        em.close();
    }
```

```
    public void alterar(Contato contato){
        EntityManager em = ConexaoJpa.getEntityManager();
        EntityTransaction tx = em.getTransaction();
        tx.begin();
        em.merge(contato);
        tx.commit();
        em.close();
    }
```

```
    public void excluir(int id){
        EntityManager em = ConexaoJpa.getEntityManager();
        EntityTransaction tx = em.getTransaction();
        tx.begin();
        Contato contato = em.find(Contato.class, id);
        em.remove(contato);
        tx.commit();
        em.close();
    }
```

```
    public Contato consultar(int id){
        EntityManager em = ConexaoJpa.getEntityManager();
        EntityTransaction tx = em.getTransaction();
        Contato contato = em.find(Contato.class, id);
        em.close();
        return contato;
    }
```

```
    public List<Contato> listar(){
        EntityManager em = ConexaoJpa.getEntityManager();
        EntityTransaction tx = em.getTransaction();
        Query query = em.createQuery("select c from Contato c");
```

```

        List<Contato> lista = query.getResultList();
        em.close();
        return lista;
    }

}

package aula09;
import java.util.List;

import javax.swing.JOptionPane;

public class TesteContatoJpa {

    public static void main(String[] args) {
        Contato contato = new Contato();
        contato.setNome("Joaquim");
        ContatoDaoJpa cdao = new ContatoDaoJpa();
        cdao.incluir(contato);
        int codigoExcluir = new
Integer(JOptionPane.showInputDialog("Qual registro excluir?"));
        cdao.excluir(codigoExcluir);

        List<Contato> lista = cdao.listar();
        for(Contato c : lista){
            System.out.printf("\n%10d%-30s",c.getCodigo(),
c.getNome());
        }
    }

}

```

```

<?xml version="1.0" encoding="UTF-8"?>
<persistence version="1.0"
    xmlns="http://java.sun.com/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd">
    <persistence-unit name="agenda">
        <class>aula09.Contato</class>

        <properties>
            <property name="hibernate.hbm2ddl.auto"
value="update" />
            <property name="hibernate.dialect"
                value="org.hibernate.dialect.PostgreSQLDialect"
/>
            <property name="hibernate.max_fetch_depth" value="3"

```

```
/>
```

```
    <!-- mostra o sql no log -->
    <property name="hibernate.show_sql" value="true" />

    <!-- dados da conexao com o banco -->
    <property name="hibernate.connection.driver_class"
        value="org.postgresql.Driver" />
    <property name="hibernate.connection.username"
        value="postgres" />
    <property name="hibernate.connection.password"
        value="123456" />
    <!-- dados da url de conexao com o banco -->
    <property name="hibernate.connection.url"
        value="jdbc:postgresql://127.0.0.1/meuBanco" />
    </properties>
</persistence-unit>
</persistence>
```

```
//Codigo Livre não entra no material de apoio
//Exemplo de consulta todos no banco
    //listar os contatos
    Query query = em.createQuery("select e from Contato e");
    List<Contato> listaContatos = query.getResultList();
    for (Contato c : listaContatos){
        System.out.printf("%20s %20s %20s", c.getNome(),
c.getEndereco(), c.getTelefone());
    }
```