

Exercício 02 – Utilização do DAO Genérico com Anotações JPA

Considere a seguinte interface:

```
import java.io.Serializable;
import java.util.List;

public interface Dao<T> {

    void adicionar(T entidade);
    List<T> listar();
    void atualizar(T entidade);
    void remover(T entidade);
    T buscar(int id);
}
```

Criar uma classe chamada JpaUtil, contendo um método para retornar uma instância de EntityManagerFactory:

```
package br.com.fiap.dao;

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

public class JpaUtil {
    private static EntityManagerFactory emf =
        Persistence.createEntityManagerFactory("jpaPU");

    public static EntityManager getEntityManager() {
        return emf.createEntityManager();
    }
}
```

Escrever uma classe chamada **GenericDao** que implemente esta interface. Nesta classe implementar os métodos da API JPA:

```
package br.com.fiap.dao;

import java.util.List;
import javax.persistence.EntityManager;

public class GenericDao<T> implements Dao<T> {

    private final Class<T> classe;
    protected EntityManager em;

    public GenericDao(Class<T> classe) {
        this.classe = classe;
    }

    @Override
    public void adicionar(T entidade) {
        em = JpaUtil.getEntityManager();
        em.getTransaction().begin();
        em.persist(entidade);
        em.getTransaction().commit();
        em.close();
    }
}
```

```

    }

    @SuppressWarnings("unchecked")
    @Override
    public List<T> listar() {
        em = JpaUtil.getEntityManager();
        return em.createQuery("From " + classe.getSimpleName()).getResultList();
    }

    @Override
    public void atualizar(T entidade) {
        em = JpaUtil.getEntityManager();
        em.getTransaction().begin();
        em.merge(entidade);
        em.getTransaction().commit();
        em.close();
    }

    @Override
    public void remover(T entidade) {
        em = JpaUtil.getEntityManager();
        em.getTransaction().begin();
        em.remove(em.merge(entidade));
        em.getTransaction().commit();
        em.close();
    }

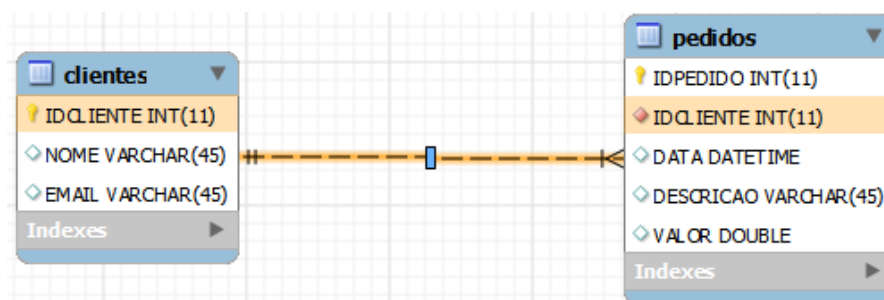
    @Override
    public T buscar(int id) {
        em = JpaUtil.getEntityManager();
        em.getTransaction().begin();
        T entidade = em.find(classe, id);
        em.getTransaction().commit();
        em.close();

        return entidade;
    }
}

```

Definir uma classe para testar os métodos. Sugestão: Na classe contendo o método main(), definir métodos locais para testar cada uma das funcionalidades.

Utilizar esta classe com o banco de dados definido no Exercício 01:



Bom trabalho a todos!

Prof. Emilio