

```
package Fidelizacao.dao.jpa;
```

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

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

```
import javax.persistence.NoResultException;
```

```
import javax.persistence.PersistenceContext;
```

```
import javax.persistence.TypedQuery;
```

```
import org.springframework.stereotype.Repository;
```

```
import org.springframework.transaction.annotation.Transactional;
```

```
import Fidelizacao.dao.DaoCliente;
```

```
import Fidelizacao.model.Cliente;
```

```
@Repository(value = "JPACliente")
```

```
public class JPACliente implements DaoCliente {
```

```
    @PersistenceContext
```

```
    private EntityManager manager;
```

```
    @Override
```

```
    public Cliente buscar(Long id) {
```

```
        return manager.find(Cliente.class, id);
```

```
    }
```

```
    @Override
```

```
    public List<Cliente> buscarTodos() {
```

```
        TypedQuery<Cliente> query = manager.createQuery("from Cliente where  
status = true order by nome", Cliente.class);
```

```
        return query.getResultList();
```

```
    }
```

```
@Override

public List<Cliente> buscarTodosInativos() {

    TypedQuery<Cliente> query = manager.createQuery("from Cliente where
status = false", Cliente.class);

    return query.getResultList();

}
```

```
@Transactional

@Override

public void inserir(Cliente obj) {

    manager.persist(obj);

}
```

```
@Transactional

@Override

public void alterar(Cliente obj) {

    manager.merge(obj);

}
```

```
@Transactional

@Override

public void desativar(Cliente obj) {

    obj.setStatus(false);

    manager.merge(obj);

}
```

```
@Transactional

@Override

public void deletar(Cliente obj) {

    manager.remove(obj);

}
```

```
}
```

```
@Override
```

```
public Cliente login(String cpf) {  
    TypedQuery<Cliente> query = manager.createQuery("from Cliente where cpf =  
:cpf and status = true",  
        Cliente.class);  
    query.setParameter("cpf", cpf);  
  
    try {  
        return query.getSingleResult();  
    } catch (NoResultException e) {  
        e.printStackTrace();  
        return null;  
    } catch (Exception e) {  
        e.printStackTrace();  
        return null;  
    }  
}
```

```
@Override
```

```
public void atualizaTokenPushNotification(Cliente cliente, String token) {  
    // TODO Auto-generated method stub  
    cliente.setTokenPushNotification(token);  
    manager.merge(cliente);  
}
```

```
@Override
```

```
public List<Cliente> buscarClientesProximoFidelizacao() {  
    TypedQuery<Cliente> query = manager.createQuery(  

```

```
        "select c from Fidelizacao f inner join f.cliente c inner join  
f.programaFidelizacao pf where (select count(*) from Fidelizacao fide where  
fide.cliente.clienteld = c.clienteld) >= (pf.qtdPremio - 3) group by c.nome",  
        Cliente.class);
```

```
    try {  
        return query.getResultList();  
    } catch (NoResultException e) {  
        e.printStackTrace();  
        return null;  
    } catch (Exception e) {  
        e.printStackTrace();  
        return null;  
    }  
}
```

```
}
```

```
}CC
```