



O JPA fornece um `EntityManager` para se **trabalhar com operações no banco de dados**, ele esconde a maioria do código pesado e se torna realmente um gerenciador de entidades simples de se utilizar.

O `EntityManager` é uma **interface** portanto sua criação é um pouco complexa, por isso usa uma factory que também é uma **interface** e depende de uma **classe concreta** chamada **Persistence**. A classe de **Persistence** tem **métodos estáticos para a criação de um EntityManager** que recebe como parâmetro o nome da **Persistence-Unit do xml**:

```
EntityManagerFactory factory = Persistence.createEntityManagerFactory("loja");
EntityManager entityManager = factory.createEntityManager();
```

```
<persistence-unit name="loja" transaction-type="RESOURCE_LOCAL">
```

Algumas **propriedades** das **implementações concretas** do JPA são muito úteis **pro dia a dia**, como a **propriedade de show_sql** que permite ver no console **as queries que são geradas**, além de uma propriedade bem útil **APENAS EM DESENVOLVIMENTO** que é a de **criação de tabelas pelo hibernate usando as classes mapeadas** (isso não é usado em produção, é aconselhável usar algum versionador de banco de dados como o flyway por exemplo):

```
<property name="hibernate.dialect" value="org.hibernate.dialect.H2Dialect"/>
<property name="hibernate.show_sql" value="true"/>
<property name="hibernate.hbm2ddl.auto" value="update"/>
</properties>
```

O `EntityManager` sendo um gerenciador de entidades, obviamente vai saber trabalhar com métodos de CRUD e até mais além, o jeito mais simples para salvar um objeto é usar o método **.persist(Objeto)**. Como estou testando com a propriedade **RESOURCE_LOCAL no persistence-unit**, eu preciso **delimitar a transação com o banco de dados (manipular ela por conta própria)** já que o controle de transação não é automático:

```
public static void main(String[] args) {
    Produto celular = new Produto();

    celular.setNome("Nokia");
    celular.setDescricao("Velho");
    celular.setPreco(new BigDecimal("10.000"));

    EntityManagerFactory factory = Persistence.createEntityManagerFactory("loja");
    EntityManager entityManager = factory.createEntityManager();

    entityManager.getTransaction().begin();
    entityManager.persist(celular);
    entityManager.getTransaction().commit();
    entityManager.close();
}
```

```
Hibernate: create table produtos (id bigint generated by default as identity, descricao varchar(255), nome varchar(255), preco decimal(19,2), primary key (id))
rev. 13, 2022 11:27:11 AM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator: InitiateService
INFO: HH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Hibernate: insert into produtos (id, descricao, nome, preco) values (null, ?, ?, ?)

Process finished with exit code 0
```