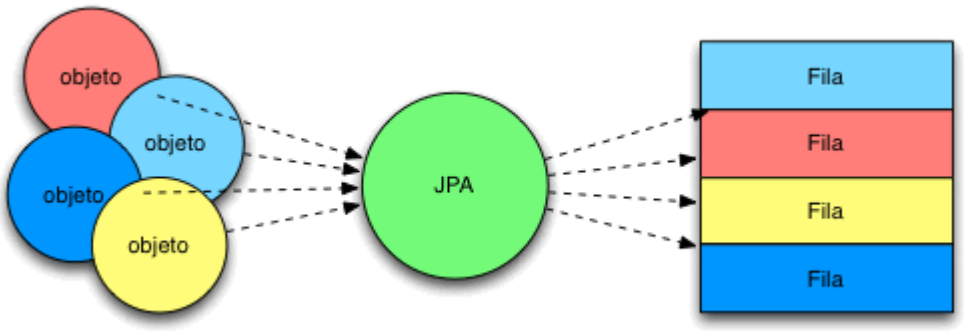
CURSO JPA

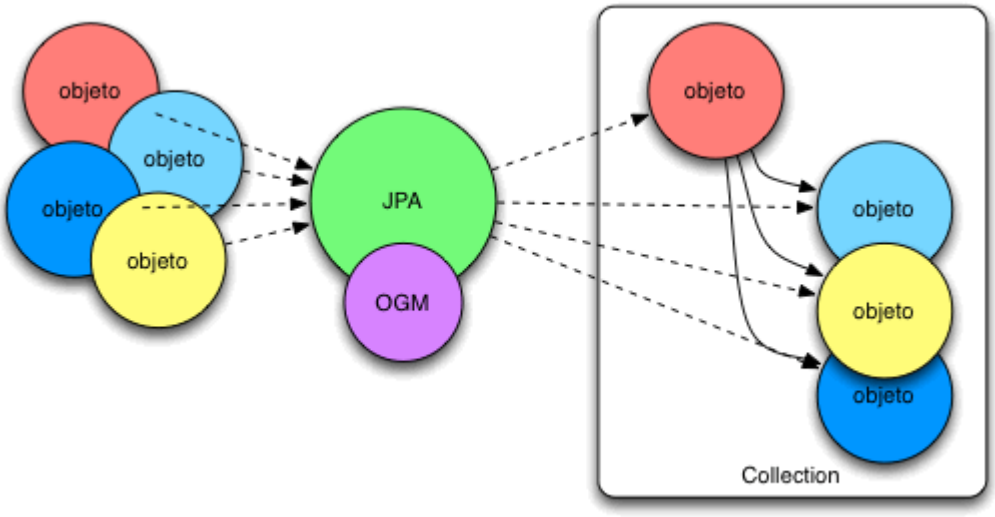
GRATIS

APUNTATE!!

El mundo de NoSQL sigue avanzando y tiene sus ventajas. JPA como standard siempre se ha centrado en facilitar la persistencia de objetos contra una base de datos Entidad-Relación tipo Oracle/MySQL/SQL Server.



Sin embargo no tenemos porque limitarnos a esto ya que JPA (Java Persistence API) puede acabar utilizandose para almacenar objetos en otros tipos de Repositorios de información uno de ellos puede ser por ejemplo una colección de objetos MongoDB.



Para ello nos vamos a apoyar en Hibernate OGM (Hibernate Object Grid Mapper) .Este framework nos permite integrar de una manera bastante transparente JPA con el mundo NoSQL.

Para integrar JPA MongoDB usaremos las siguientes dependencias de Maven:


```
&lt;/p&gt;<br>
&lt;p&gt;&amp;amp;lt;dependency&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;groupId&amp;amp;gt;org.hibernate.ogm&amp;amp;lt;/groupId&a
mp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;artifactId&amp;amp;gt;hibernate-ogm-
mongodb&amp;amp;lt;/artifactId&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;version&amp;amp;gt;4.1.0.Beta8&amp;amp;lt;/version&amp;amp
;gt;&lt;br&gt;<br>
&amp;amp;lt;/dependency&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;dependency&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;groupId&amp;amp;gt;org.hibernate.javax.persistence&amp;amp
;lt;/groupId&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;artifactId&amp;amp;gt;hibernate-jpa-2.0-
api&amp;amp;lt;/artifactId&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;version&amp;amp;gt;1.0.1.Final&amp;amp;lt;/version&amp;amp
;gt;&lt;br&gt;<br>
&amp;amp;lt;/dependency&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;dependency&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;groupId&amp;amp;gt;org.jboss.spec.javax.transaction&amp;am
p;lt;/groupId&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;artifactId&amp;amp;gt;jboss-transaction-
api_1.1_spec&amp;amp;lt;/artifactId&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;version&amp;amp;gt;1.0.0.Final&amp;amp;lt;/version&amp;amp
;gt;&lt;br&gt;<br>
&amp;amp;lt;scope&amp;amp;gt;provided&amp;amp;lt;/scope&amp;amp;gt;&lt
;br&gt;<br>
&amp;amp;lt;/dependency&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;dependency&amp;amp;gt;&lt;br&gt;<br>
```

<groupId>org.jboss.jbossts</groupId>

<artifactId>jbossjta</artifactId>

<version>4.16.4.Final</version>

</dependency></p>
<p>

Una vez tenemos las dependencias definidas tendremos como siempre que configurar un fichero persistence.xml que nos permite conectar con la base de datos.

</p>
<p><persistence
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_2_0.xsd"

version="2.0">

<persistence-unit name="UnidadOGM" transaction-type="JTA"></p>
<p><provider>org.hibernate.ogm.jpa.HibernateOgmPersistence</provider>

<class>com.arquitecturajava.Persona</class>

<properties></p>
<p><property name="hibernate.ogm.datastore.provider" value="mongodb" />

<property
name="hibernate.ogm.datastore.grid_dialect"


```
value="org.hibernate.ogm.datastore.mongodb.MongoDBDialect"
/&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;property name="hibernate.ogm.datastore.database"
value="arquitecturajava" /&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;property name="hibernate.ogm.mongodb.host"
value="127.0.0.1" /&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;property name="hibernate.ogm.mongodb.port" value="27017"
/&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;/properties&amp;amp;gt;&lt;br&gt;<br>
&amp;amp;lt;/persistence-unit&amp;amp;gt;&lt;/p&gt;<br>
&lt;p&gt;&amp;amp;lt;/persistence&amp;amp;gt;&lt;/p&gt;<br>
&lt;p&gt;
```

La configuración es relativamente sencilla simplemente cambiamos el proveedor habitual a org.hibernate.ogm.jpa.HibernateOgmPersistence y configuramos el dialecto para MongoDB (MongoDBDialect). Una vez hecho esto creamos una clase Persona y un programa Principal.

```
&lt;/p&gt;<br>
&lt;p&gt;package com.arquitecturajava.negocio;&lt;/p&gt;<br>
&lt;p&gt;import javax.persistence.Entity;&lt;br&gt;<br>
import javax.persistence.Id;&lt;/p&gt;<br>
&lt;p&gt;import org.hibernate.annotations.Type;&lt;/p&gt;<br>
&lt;p&gt;@Entity&lt;br&gt;<br>
public class Persona {&lt;/p&gt;<br>
&lt;p&gt;@Id&lt;br&gt;<br>
@Type(type = "objectid")&lt;br&gt;<br>
private String id;&lt;/p&gt;<br>
&lt;p&gt;public String getId() {&lt;br&gt;<br>
return id;&lt;br&gt;<br>
}&lt;/p&gt;<br>
```

```
&lt;p&gt;public void setId(String id) {&lt;br&gt;<br>
this.id = id;&lt;br&gt;<br>
}&lt;/p&gt;<br>
&lt;p&gt;private String nombre;&lt;br&gt;<br>
private String apellidos;&lt;/p&gt;<br>
&lt;p&gt;public String getApellidos() {&lt;br&gt;<br>
return apellidos;&lt;br&gt;<br>
}&lt;/p&gt;<br>
&lt;p&gt;public void setApellidos(String apellidos) {&lt;br&gt;<br>
this.apellidos = apellidos;&lt;br&gt;<br>
}&lt;/p&gt;<br>
&lt;p&gt;public String getNombre() {&lt;br&gt;<br>
return nombre;&lt;br&gt;<br>
}&lt;/p&gt;<br>
&lt;p&gt;public Persona() {&lt;br&gt;<br>
super();&lt;br&gt;<br>
}&lt;/p&gt;<br>
&lt;p&gt;public Persona(String nombre) {&lt;br&gt;<br>
super();&lt;br&gt;<br>
this.nombre = nombre;&lt;br&gt;<br>
}&lt;/p&gt;<br>
&lt;p&gt;public void setNombre(String nombre) {&lt;br&gt;<br>
this.nombre = nombre;&lt;br&gt;<br>
}&lt;/p&gt;<br>
&lt;p&gt;}&lt;/p&gt;<br>
&lt;p&gt;


&lt;/p&gt;<br>
&lt;p&gt;&amp;amp;amp;nbsp;&lt;/p&gt;<br>
&lt;p&gt;
```

```
&lt;/p&gt;<br>
&lt;p&gt;package com.arquitecturajava.negocio;&lt;/p&gt;<br>
&lt;p&gt;import java.util.List;&lt;/p&gt;<br>
&lt;p&gt;import javax.persistence.EntityManager;&lt;br&gt;<br>
import javax.persistence.EntityManagerFactory;&lt;br&gt;<br>
import javax.persistence.Persistence;&lt;/p&gt;<br>
&lt;p&gt;public class Principal {&lt;/p&gt;<br>
&lt;p&gt;public static void main(String[] args) {&lt;/p&gt;<br>
&lt;p&gt;EntityManagerFactory emf = Persistence&lt;br&gt;<br>
.createEntityManagerFactory("UnidadOGM");&lt;/p&gt;<br>
&lt;p&gt;System.out.println(emf);&lt;br&gt;<br>
EntityManager em = emf.createEntityManager();&lt;br&gt;<br>
List&amp;amp;lt;Persona&amp;amp;gt; lista=em.createQuery("select p
from Persona p",Persona.class).getResultList();&lt;/p&gt;<br>
&lt;p&gt;for(Persona p :lista) {&lt;/p&gt;<br>
&lt;p&gt;System.out.println(p.getNombre());&lt;br&gt;<br>
System.out.println(p.getApellidos());&lt;br&gt;<br>
}&lt;br&gt;<br>
em.close();&lt;/p&gt;<br>
&lt;p&gt;}&lt;/p&gt;<br>
&lt;p&gt;}&lt;/p&gt;<br>
&lt;p&gt;
```

El código es idéntico a un código de JPA . Con la salvedad de que hemos tenido que usar una anotación especifica de Hibernate para poder mapear el típico Object Id de MongoDB.

Los datos que seleccionamos de la base de datos Mongo saldrán por pantalla.

```
INFO: OGM000001: Hibernate OGM 4.1.0.Beta8
org.hibernate.ogm.jpa.impl.OgmEntityManagerFactory@45d05
pedro
gomez
ana
gomez
ana
perez
```

Tenemos que estar cada día mas atentos a este proyecto de Hibernate que tiene un gran futuro y nos permitirá integrar de forma más sencilla nuestras clases Java con Repositorios de datos heterogéneos.

Otros artículos relacionados: Introducción a JPA , JPA y relaciones ,Libro de JPA