

Todos hemos trabajado con Java y JDBC para acceder a las distintas bases de datos SQL (Oracle,SQLServer,MySQL). Poco a poco tendremos que irnos haciendo a la idea de comenzar a trabajar con bases de datos NoSQL. Uno de los casos más habituales sera usar Java y MongoDB como base de datos documental. Conectarse a esta base de datos es relativamente fácil ya que disponemos de un conjunto de abstracciones bastante sencillas de entender. Lo primero que vamos a hacer es incluir una dependencia de Maven hacia el driver de Mongo en nuestro proyecto.

```
<dependency>
<groupId>org.mongodb</groupId>
<artifactId>mongo-java-driver</artifactId>
<version>2.12.4</version>
</dependency>
```

Una vez tenemos el driver asignado el siguiente paso es crear un programa sencillo que se conecte a la base de datos de Mongo.

```
import com.mongodb.DB;
import com.mongodb.DBCollection;
import com.mongodb.MongoClient;

public class Principal {

public static void main(String[] args) {
```

```
try {
MongoClient mongo= new MongoClient();
DB db= mongo.getDB("Prueba");

DBCollection coleccion= db.getCollection("Persona");

BasicDBObject objeto= new BasicDBObject();

Persona p= new Persona("pepe","perez");
objeto.put("nombre",p.getNombre() );
objeto.put("apellidos",p.getApellidos());
coleccion.insert(objeto);
} catch (UnknownHostException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
}
```

El programa es muy sencillo pero al ser todos conceptos nuevos vamos a desglosarlo.

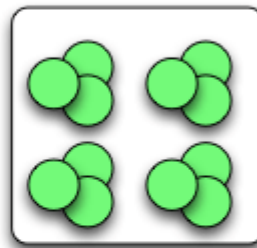
```
MongoClient mongo=  
new MongoClient("127.0.0.1");  
DB db= mongo.getDB("Prueba");
```

1

Base  
Datos

```
DBCollection coleccion=  
db.getCollection("Persona");
```

2



Colección

```
BasicDBObject objeto=  
new BasicDBObject();  
Persona p=  
new Persona("pepe", "perez");  
objeto.put("nombre", p.getNombre() );  
objeto.put("apellidos", p.getApellidos());  
coleccion.insert(objeto);
```

3



Documento

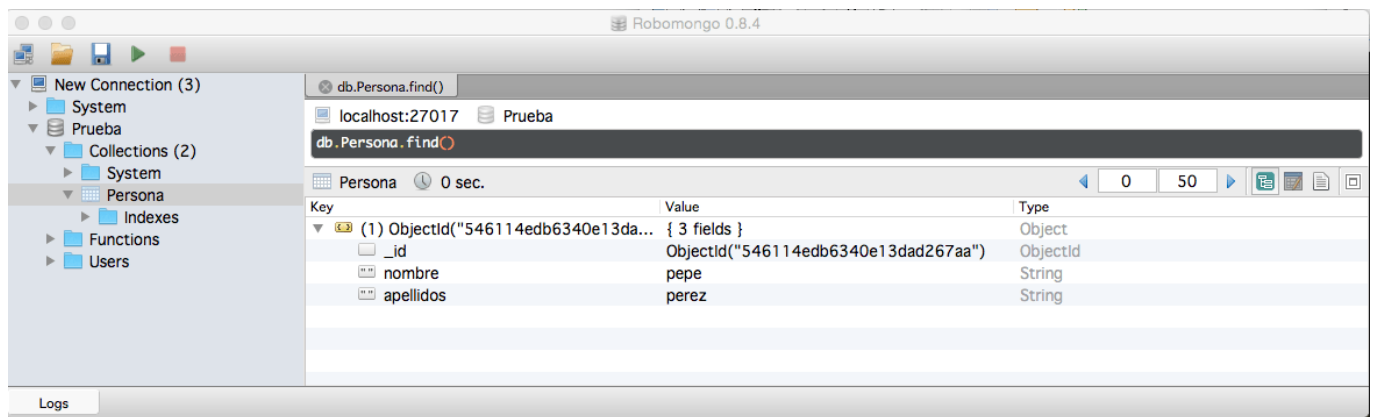
El programa se divide en tres pasos:

1. En el primer paso usamos un cliente para conectarnos a Mongo desde Java. No necesitamos pasar el puerto o el servidor como parámetros porque se trata de algo local y usa los parámetros por defecto.
2. El segundo paso creamos una colección que no es ni más ni menos que un grupo de documentos con los que vamos a trabajar. En nuestro caso la colección es de Personas.
3. En el último paso creamos un objeto persona y pasamos sus parámetros a un BasicDBObject

que hace referencia a un documento concreto. Realizada esta última operación insertamos en la base de datos el registro.

## Java MongoDB y RoboMongo

Una de las herramientas más habituales para trabajar con MongoDB es **RoboMongo** que nos permite acceder al servidor desde un interface amigable.



Esto nos sirve de primera toma de contacto con Java y MongoDB. Pero es fácil darse cuenta que estamos accediendo de una forma muy similar a JDBC. En futuros artículos nos orientaremos más hacia el mundo de JPA e Hibernate donde el nivel de abstracción es superior.

Otros artículos relacionados: [Ejemplo JPA](#) , [Java y JSON](#) , [NoSQL](#)