Item 4 – Requirement A+

The implementation of this requirement has been done following the next steps:

1. We have looked for some information about the technology to use on Hibernate official webpage.
2. After the previous step, we have put the neccessary dependency in our pom.xml.

```
<dependency>
<groupId>org.hibernate</groupId>
<artifactId>hibernate-search</artifactId>
<version>5.0.1.Final</version>
</dependency>
```

3. Then, we have modified the FixUpTask class. In which, we have added the annotations '@Indexed' at the top and '@Field' to every getter that we wanted to work with.
4. Later, we have made a class in package '' called by 'Utiles' with the official code given by Hibernate. Due to some limitations of the technology, we have had to modify the persistence.xml file with the following lines in order to get our goal:

```
<property name="hibernate.search.default.directory_provider" value="filesystem"/>
<property name="hibernate.search.default.indexBase"
value="/var/lucene/indexes"/>
```

5. Finally, we have used the provided class named by 'SchemaPrinter' for the showing of the possible results returned by any desired query.

The functionality consists on:

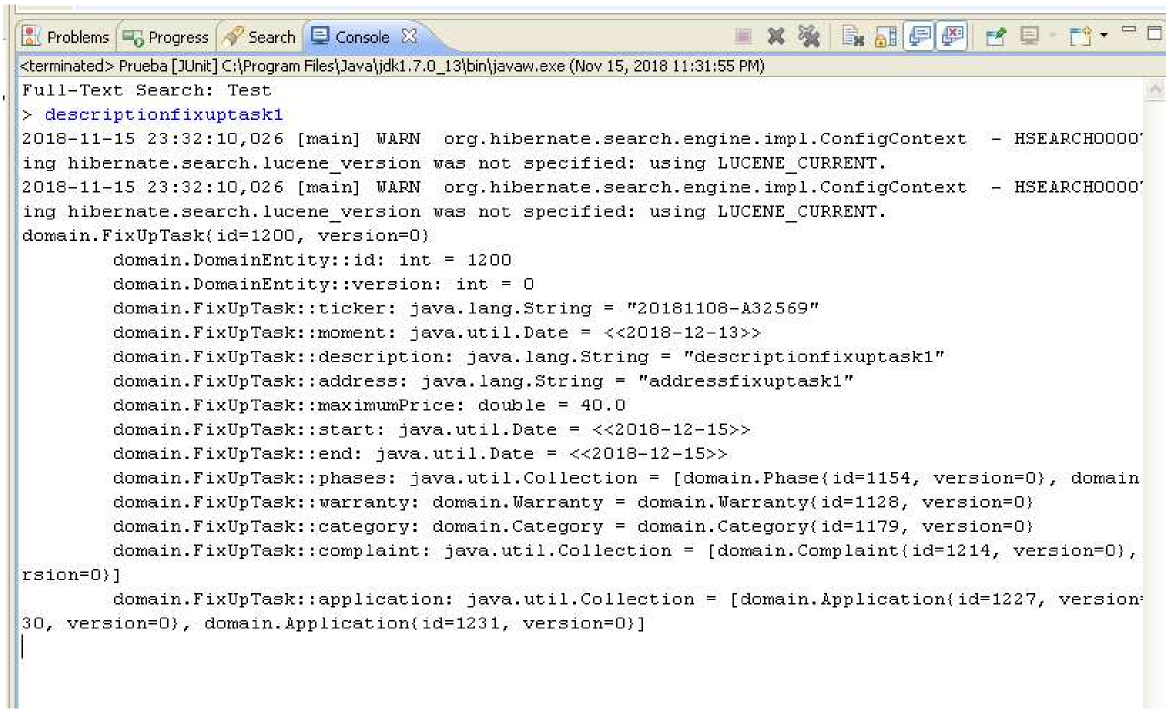
ConsoleReader reads the key words that the user introduces by console. Then, it's turned to a string which is taken as parameter into an auxiliar method that contains the neccesary code for the search.

First of all, we make an 'Entity Manager' instance that makes a search to the persistence unit with the string introduced. Then, we call to 'QueryBuilder' class in order to make our solution with class mentioned before, FixUpTask.

Down, you find a 'Query - luceneQuery' class in which we introduce the fields that are going to be checked for our consultation. Now, we let to Javax Query takes the object 'Query' created before with the class used.

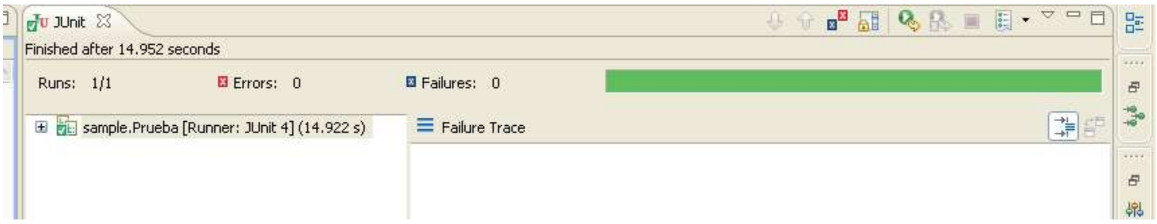
A result set is returned and given as parameter to SchemaPrinter class. This will let us to show in console the final result.

In the main method, it can be seen how a line is read and the function described is used for the experiment.



<terminated> Prueba [JUnit] C:\Program Files\Java\jdk1.7.0_13\bin\javaw.exe (Nov 15, 2018 11:31:55 PM)

```
Full-Text Search: Test
> descriptionfixuptask1
2018-11-15 23:32:10,026 [main] WARN  org.hibernate.search.engine.impl.ConfigContext  - HSEARCH0000
ing hibernate.search.lucene_version was not specified: using LUCENE_CURRENT.
2018-11-15 23:32:10,026 [main] WARN  org.hibernate.search.engine.impl.ConfigContext  - HSEARCH0000
ing hibernate.search.lucene_version was not specified: using LUCENE_CURRENT.
domain.FixUpTask{id=1200, version=0}
        domain.DomainEntity::id: int = 1200
        domain.DomainEntity::version: int = 0
        domain.FixUpTask::ticker: java.lang.String = "20181108-A32569"
        domain.FixUpTask::moment: java.util.Date = <<2018-12-13>>
        domain.FixUpTask::description: java.lang.String = "descriptionfixuptask1"
        domain.FixUpTask::address: java.lang.String = "addressfixuptask1"
        domain.FixUpTask::maximumPrice: double = 40.0
        domain.FixUpTask::start: java.util.Date = <<2018-12-15>>
        domain.FixUpTask::end: java.util.Date = <<2018-12-15>>
        domain.FixUpTask::phases: java.util.Collection = [domain.Phase{id=1154, version=0}, domain
        domain.FixUpTask::warranty: domain.Warranty = domain.Warranty{id=1128, version=0}
        domain.FixUpTask::category: domain.Category = domain.Category{id=1179, version=0}
        domain.FixUpTask::complaint: java.util.Collection = [domain.Complaint{id=1214, version=0},
rsion=0}]
        domain.FixUpTask::application: java.util.Collection = [domain.Application{id=1227, version
30, version=0}, domain.Application{id=1231, version=0}]
```



Finished after 14.952 seconds

Runs: 1/1   Errors: 0   Failures: 0

sample.Prueba [Runner: JUnit 4] (14.922 s)