# EJERCICIO

In order to implement this exam, the following files are provided:

* Passwords.txt: A text file with the 10,000 most common passwords used over the Internet, one password per line.
* Methods.java: Code of several methods that count different types of characters on a String. It is also included a sample *password strength* formula, that return a number (int) that indicates how good that password is (the greater the better).

We need to develop an application to read, sort and show the 10,000 most common passwords used over the Internet. We also need to store them in a compressed format once shown. For that purpose, develop that application following these guidelines:

1. Create a FileHandler class that is able to read each line of the file into an ArrayList of Strings (one element per line). In order to overcome “file not found” exceptions when reading the password file, it is recommended to place this file on the project folder of the workspace. For example, if the project is called “exam2” and the Eclipse workspace resides in C:\Users\workspace, you should place the file in

c:\Users\workspace\exam2\Passwords.txt.

1. Create a Password class able to store one password (String). Add to this class all the methods provided in Methods.java, so all of them work with the class’ stored password instead of with one String parameter. If you need to remove parameters from the methods or make some alterations to its code, you can do it freely. Once this class is created, create a Java ArrayList of 10,000 Password objects with each one of the 10,000 passwords read in step 1

1. Implement the Comparable interface so passwords can be sorted by its length. Sort all the passwords and show the longest one to the user.
2. Create a Comparator that is able to sort the passwords using its *strength score*. Sort all the passwords and show the strongest one to the user.
3. Write the result of calling the toString method of each Password object into a Zip file called passwords.zip. Place this method on the FileHandler class. The toString method of the Password class must follow the following format: <password>:<strength score> (Ej.: films+pic+galeries:124**)**
4. Develop unit tests to test the *strength score* calculations and both sorting procedures. To do that, consider that the longest and also the most complex password present in the file is “films+pic+galeries” and its strength score is 124

**Use exceptions on every part of the exercise you think they should be placed.**