# Summary exercise

# 1. Motivation

Throughout the course we have learned different techniques, procedures and resources that help us in our task as programmers.

Now it's time to put it all together and practice all that we have learned in this course.

# 2. Your work

You're gonna think and deploy a simple but orginal project (at least 3 classes, related between them).

The steps that you have to follow are:

1. **Think about the project**, the classes that it will use, the relationship between them and the functionalities that the project will have. You must send an email to the teacher with the following information and wait for his confirmation or possible modifications before continuing:

   - **Project's name**
   - **Short description**: a sentence about the objective or the functionality that the project will solve.
   - **Technical remarks:** Name of the classes and their relationships (), as well as any technical issue that justifies their complexity (reading from files, use of arrays or collections, inheritance). It is not necessary to take up more than one paragraph.

   Once you have received confirmation from the teacher, you can continue with the second point.

2. **Write all the requirements of your project** as detailed as possible (in the style of the statements of the exercises that we have used in the programming module), with details about the classes, their relationships, their attributes, methods, etc. As well as any design indication that you consider. You can do this point simultaneously with the next one if it is easier for you.

3. **Design the entire project in argoUML**, remember that once designed and approved this design will generate the source code headers of your future application. The design must contain:

   1. **Diagram of all classes**, with their attributes and methods.
   2. **Case use diagram**: The context diagram and the global diagram with all the use cases.

   Wait for the teacher's confirmation before moving on to point 4.

4. Create a **repository on gitHub**, it can be private if you add the teacher as a collaborator, or public. Send the link to the teacher and make sure he can access it.

   You must use this repository to generate and update your project, here you must post the project requirements (for example in the README.md), as well as the UML diagrams both in PDF and in .zargo. You must use this repository from the beginning, and not use it at the end to push everything after it is finished. The project must have different commits in which the evolution of the project can be seen.

5. Remember to use **JavaDoc** to document **all** classes, methods, etc.

6. [OPTIONAL +1 point] Include in your Project Unit tests for the classes at least for 3 methods of each class.

7. Generate a document (**final summary**, no more than one sheet is necessary) with your opinion on the part of the whole process that you liked the most, and also the part that you liked the least and why. Specify in the document how the project could be extended to become a real application, indicating what functionalities you would like to add to it

(although you are not yet able to do so). Include in the document the link to the gitHub repository and make sure it is visible by the teacher for evaluation.

> Anything that is not hosted in the repository will not be taken into account.

8. Send the **PDF** document through the **AULES** platform to the teacher.