

Guided Exercise 2

Statement

Collective Code Ownership and Coding Standards



Software Development

February 2021

Grado en Ingeniería Informática - Doble Grado en Informática y ADE

TABLE OF CONTENTS

OBJECTIVES	3
DESCRIPTION OF THE COMPONENT	3
TASKS TO DO	4
RULES AND PROCEDURES	6
EVALUATION	7

OBJECTIVES

The purposes of this guided exercise are:

- Learn the best practices of collective code ownership to be applied during the development of a project, following agile methodologies' principles.
- Establish a coding standard, accepted and implemented by the entire team, to ensure that all the project developers work in a coordinated way.
- Apply that coding standard using a tool in the integrated development Environment.

DESCRIPTION OF THE COMPONENT

UC3MLogistics is a company specialising in developing computer technologies for order management. He is currently developing a warehouse product management system and controlling the dates on which the products will be processed and delivered. One of the requirements is to handle product codes.

UC3MLogistics have decided to implement a component for validating a product code that could have been read, for example, through a barcode. These codes will follow the EAN-13 standard:

https://en.wikipedia.org/wiki/International_Article_Number

The UC3MLogistics software engineers began implementing this component, but given its difficulties, they decided to launch a development contract with a group of independent software engineers (you).

The management system must provide the necessary mechanisms to generate unique order identifiers from input data provided by a text file in JSON format, which will serve

to register orders in the system. You can download the first version of the component from Aula Global. You must include the code in a PyCharm project and implement a new method for checking the validity of a given EAN-13 code.

This component must be developed in Python 3.8, using the IDE PyCharm and configuring a virtual environment for the project. The code must be shared in a private repository on GitHub.

TASKS TO DO

1. Configure the PyCharm environment with the necessary components to connect with GitHub and review the code with the Pylint plugin.
2. Create a private repository on GitHub for this guided exercise, following the steps outlined in the annexe file. The repository must be created by a team member and must be shared with the other team member and with the lab teacher.
3. Create a project in PyCharm, configure the virtual environment, and include the code available in Aula Global.
4. The products will be identified by EAN-13 codes:

https://en.wikipedia.org/wiki/International_Article_Number

Include a function that, given an EAN-13 code, checks whether it is correct and generates a jpeg file with the barcode. To do so, you will use the library python-barcode. In case of invalid EAN-13 codes, the component will raise an exception. You must include two examples, one with a valid EAN-13 code and the other with an invalid one.

5. Define a coding standard for your team. You must describe your coding standard in a pdf document. The appearance of this document will be taken into account.
6. Configure the Pylint plugin to recognize between 10 and 15 rules of your coding

standard. We will take into account the inclusion of more rules.

7. Eliminate all warnings and errors that Pylint detects in your code.
8. Share the project via GitHub with your partner. The two team members must have downloaded the repository, configured the project and the virtual environment, made some modifications, and submitted it at least once. Although this step is at the end, the interaction with GitHub should be continuous from the beginning of the guided exercise. If we notice that one of the members of the team has not sent any code (or the changes sent are insignificant, like dropping or commenting a line), that student may be penalized. Interaction between team members through GitHub will be taken into account: include several code exchanges between the team members with significant changes in the code.
9. Share the project via Github with the teacher. **It is important to follow the exact nomenclature of the project name defined in the annexe so** that the teacher can find your exercise when evaluating it.

The annexes of the guided exercise describe step by step the work to do. In many cases, the configuration we include is quite similar to the required one. Still, in some cases, names and other elements have been included as examples. Keep in mind that you must adjust the steps to your guided exercise's requirements.

RULES AND PROCEDURES

This exercise will be completed in pairs. Students who do not find a partner or who cannot attend the sessions should contact the corresponding professor of the subject to report their situation.

The delivery will consist of the following parts:

1) Collective code ownership.

To satisfy the submission requirements, the private GitHub repository created by each group must include the project **UC3MLogistic** with the code that correctly implements the restrictions of the coding standard defined by the students. This repository will be shared with the corresponding practice teacher.

2) Coding standard

The coding standard will be provided in a document in PDF format named **“CodingStandard.pdf”**. In each section of the coding standard, you have to include the pylint rules that implement your restrictions. This document must be included in the root directory of the software project corresponding to the requested module for each group of practices.

Likewise, this coding standard must be implemented in Pylint for PyCharm, leaving the file **.pylintrc** in the root directory. The **.pylintrc** file will include comments for identifying the changes made by the students.

You must take a screenshot of the Pylint console before modifying the code (the errors are displayed) and another screenshot after solving all the coding issues raised by Pylint (the errors should have disappeared). These screenshots will also be included in the root of the project.

Likewise, the source code found in the GitHub repository must comply with the coding standard defined by the group.

The deadline for the delivery of this guided exercise is set to Feb 24th before 23:59.

In accordance with the standards of continuous assessment established in this subject, if a team does not send the solution of the exercise before the deadline, the exercise will be evaluated with a score of 0 points.

Suggestion: Each student should keep a copy of the solution delivered until the end of the subject.

EVALUATION

The following criteria will be taken into account:

- The project is well-created and well-configured on Github, and we can clone the project without problems.
- The documents are uploaded in the project as indicated in the rules and procedures section.
- Both participants have significantly interacted with the GitHub repository.
- The coding standard has a formal document with a professional format, well structured. The document contains explanations for each aspect of the coding standard (textually describe each rule, include examples,...). The document must include the pylint rules configured in the corresponding sections. The inclusion of additional code in the project to apply the rules of the coding standard will be positively valued.
- The number of rules included is appropriate and different, showing the power of a

coding standard.

- Modified or added rules are marked in the `.pylintrc` file using comments.
- The code has been implemented and works correctly. Remember to include two examples, one with a valid EAN-13 code and the other with an invalid EAN-13 code, to demonstrate that your program works correctly.
- All the code complies with the regulations defined by the group, and there are no pylint errors or warnings.